MASTER THESIS

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Applied Sciences Technikum Wien
Degree Program Information Systems Management

Applying Machine Learning Methods for Housing Price Prediction

By: Bc. Filip Dzuroska
Student Number: 1510302098

Supervisor 1: Prof. Dr. Yevgeni Koucheryavy
Supervisor 2: FH-Prof. Dipl.-Ing. Helmut Gollner

Vienna, 19.05.2018

FH University of Applied Sciences
TECHNIKUM WIEN

Declaration of Authenticity

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz/ Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.”

Los Angeles, 19.05.2018

Place, Date

Signature

FACHHOCHSCHULE
TECHNIKUM WIEN

Kurzfassung

Der Immobilienmarkt ist eine der meist beachtesten Branchen der Weltwirtschaft. In der Zeit des ständigen Wandels, in der Immobilienpreise schneller steigen als Gehälter, suchen die Menschen nach einer Möglichkeit eine bessere Leistung für das selbe Geld zu erhalten. Dieser Ansatz wird in vielen Lebensbereichen angewandt. Das Verständnis über preisetreibende Faktoren ist für viele Branchen von entscheidender Bedeutung und Immobilien sind dabei keine Ausnahme. Die Anwendung verschiedener Methoden könnte zu einem besseren Verständnis der zugrunde liegenden Wertfaktoren von Wohnraum beitragen. Durch die Diversifizierung von Wirtschaftsdienstleistungen und -produkten können wir auch disruptive Konzepte in diesem Bereich beobachten. Einer von ihnen ist Airbnb - als Flaggschiff der Sharing Economy. Diese Masterarbeit hat als Ziel, die am besten geeigneten Techniken des maschinellen Lernens anzuwenden, um Merkmale und Werttreiber zu erfassen, die hinter dem Preis von Airbnb-Angeboten in Los Angeles stehen. Die Studie konzentriert sich auf explorative Datenanalyse, Feature-Engineering und prädiktive Modellierung unter Verwendung von Regressionsmethoden.

Schlagwörter: Machine Learning, Preisvorhersage, Unterkünfte, Wohnraum, Airbnb

Abstract

Real estate market was and still is one of the most attention-centered industries in global economy. In the era of constant change, where prices of properties are growing faster than salaries, people are looking for the smart way how to get the most for less. This approach is applied in many areas of life. To understand drivers of price is crucial for many industries and real estate is not an exception. Applying different methods might bring better understanding for underlying factors of value behind housing. With the diversification of economy services and products we can also observe disruptive concepts in this field. One of them is definitely Airbnb - as a flagship of sharing economy. In our research we are aiming to apply most appropriate machine learning techniques to comprehend features and value drivers behind the price of Airbnb listings in Los Angeles. The study is focused on exploratory data analysis, feature engineering and predictive modeling with the utilization of regression methods.

Keywords: Machine Learning, Price Prediction, Housing, Airbnb

Acknowledgements

In the first row I would like to thank my family for continuous encouragement and inspiration they gave me to follow my dreams during my studies; This pathway would not have been possible without them around me. I would like to dedicate this master thesis to my grandparents, who would be definitely proud if they could be here.

I would also like to thank to my supervisors D.Sc Yevgeni Koucheryavy for and FH-Prof. Dipl.-Ing. Helmut Gollner for constant support and professional advices throughout my master studies. It would not be possible to experience such extraordinary academic experience at UCLA without your help.

Last but not least I would like to thank to both university administration offices for their persistent efforts to make our studies professional as well as enjoyable.

Table of Contents

1 Introduction

Real estate market is one of the most significant industries in global economy. Estimation of world GDP of property sales and renting is between $75 trillion to $90 trillion [1]. In United States it is a leading sector, resulting for 2,265.7 billion dollars, which is 13.0% GDP share [2]. In the past recent years real estate market underwent major changes. Since the crisis hit the global economy, roughly 10 years have passed. In 2006 house prices were the highest in the history, before the housing bubble burst. Followed by recession which started in December 2007, there has been critical drop in real estate prices. Supply was higher than demand and the market had to go through difficult situation. A lot of people lost their homes and population's viability has dramatically declined. By July 2008, housing prices had dropped in 24 out of 25 USA metropolitan areas, with the greatest effect in California and the coast regions. [3].

Since the Great Depression, as the crisis was called, the market experienced general correction and recovery. Even the prices started to increase and has almost gained all their losses after market crash, there were several changes left as the outcomes of the crisis. The mindset of people has been affected and many started to reconsider if to own a property or rent one. In year 2006, before the whole crash, there were 36.1% renters from the whole U.S population. Until year 2014 the percentage grew approximately 5% to 41.1% [4]. There might be more factors influencing the current approach to housing like affordability and inability to buy a property or lifestyle of millennials.

But the situation was also responsible for number of market disruptions from global perspective. There has been accelerating growth and rapid changes in many sectors since then. Hand in hand with the crisis the concept of sharing economy has appeared on public and has got into attention of many. The sharing economy can be defined as “peer-to-peer activity of gaining, granting, or sharing the access to goods and services, coordinated through community-based online services” [5]. This disruptive concept has brought to public an alternative platform, known as peer-to-peer marketplace. Housing and property renting was also affected by mentioned innovation. Apartment sharing startups has received a lot of success and the concept gained on popularity. The most remarkable platform worth mentioning is Airbnb. The potential renters might browse accommodation based on their desired location, with specific requirements for amenities and in desired price range.

Airbnb was established exactly in 2008, when the crisis was peaking and since that time the platform has hosted more than 300 million people in more than 81,000 cities and 190 countries over the globe. The community is still expanding, and the popularity of the services is growing. With the massive number of 4.5 million listing, Airbnb is leading platform in this segment. New Year's Eve 2017 was the most booked date in the Airbnb history, with more

than 3 million people using the platform [6]. The company is creating a connection between homeowners, who own underutilized space on one side and guests, who are looking for a convenient accommodation in desired location for affordable price on the other side. Joining the community is for free, whether the individual would like to offer an apartment for rent or would like to rent one.

Renting an empty space might represent an eminent source of income for hosts as well as financial benefit for guests. Correct pricing of the offered property is a crucial for homes to be rented. There are several factors which influence the price. To start with, the price definitely varies based on the location of the property, type of apartment, if the guest is renting whole place, room or shared space, amenities or dates or arrival. High importance must be also taken on reviews. Every potential customer considers previous experience of the visitors. Guests as well as hosts can write a review for the property or visitors to let know a future counterparty about their experience and satisfaction either with guest or accommodation itself.

As with any type of service, understanding the pricing of the offers is important for every involved party. Dynamics of the platform, growing community, increasing competition as well as demand for offered services require smooth process with adequate property pricing to maximally leverage from sharing economy. Unlike other peer-to-peer services as Uber or Lyft, Airbnb let the pricing of the property on hosts [7]. In the current situation, homeowners might have difficulties to find the appropriate price. Airbnb has also found this fact as a weak point and spent a lot of sources to develop a system for price recommendation from supply point of view. In this dynamic pricing functionality, system present to hosts the probability of property booking based on desired homeowner price [8].

In our study we aim to conduct a research of the Airbnb listings in metropolitan area to get knowledge and understanding of specific property features and develop models for proper price prediction for hosts and guests. Applying machine learning methods on price prediction is a hot area of research and right utilization of algorithm might bring service providers as well as customers huge benefits in personalized services. Win-win situation should be an aim of every service provider and for Airbnb we believe that there are more factors affecting price of real estate. Number of local services, social factors or population construction might also increase performance and correct price estimates for space offering. The final objective of the research is to bring better understanding behind listings price by application of machine learning methods.

1.1 Problem Statement

Research of predicting models and their future outcomes is in agenda of many professional. When it comes to real estate market, attention is put on housing price prediction. Every single company is trying to get ahead of competition, secure leading position on the market, minimize costs and maximize profits. The same applies for Airbnb, a young company, providing platform for accommodation, where hosts or users might offer their home for rent. The company gets a provision from every realized booking in for of service fee and this fact drives Airbnb to make as much bookings as possible.

With this in mind, company developed a special model for listing price estimation, called Airbnb Smart Pricing, which aims to support hosts in correct price setting for their offer. But many hosts complain that recommended price is lower as it should be. Hosts would rather prefer accurate prices with, with profit maximization not only on side of Airbnb but on side of hosts themselves. Other side of Airbnb users is composed by guests or visitors, who would like to rent an accommodation but doesn't know accurate price which they should pay. It would be very useful to have understanding if host is not overpricing an accommodation. To satisfy both groups, it is necessary to develop an independent model with accurate information of home much is listing worth.

1.2 Objectives

The objective of the study is to conduct a research in the field of machine learning and price prediction of Airbnb listings for specific metropolitan area based on chosen features. The results of the first stage will be an analysis of related works and solutions. Subsequently research will focus on a detailed description of the optimal machine learning methods and their potential application for price prediction from theoretical point of view. Aim of a practical part is to conduct an analysis of available data set, preprocessing and preparation of data, application of chosen machine learning methods and model development. As a final result of the research will be a working price prediction model for Airbnb listings.

The following research question and sub questions are posed:

- What are the existing systems for Housing Price Estimation?
- What kind of machine learning methods are best suitable for price prediction of Airbnb listings?
- What are the limitations of existing systems?
- What is the performance of the existing systems?
- How to develop good performing model?

- What features are crucial for the development of the accurate model for price prediction of Airbnb listings in LA?
- What other open source features might be utilized for model prediction?

Stated questions serve as the fundamental part of the study and will be answered within the scope of the research. Following activities are not in the scope of the research:

- Model integration – research will include model development, testing and evaluation and might serve in the future as a reference model for the integration within price prediction system architecture of Airbnb listings or similar platform, but in the current research it is excluded.

1.3 Methodological Consideration and Expected Results

Based on the research scope, objectives and questions following methods are in the scope of the research for its successful finalization:

- Resources research, review and analysis – collect articles, journals, research papers, online materials or books related to our study, conduct detailed analysis of literature and available materials.
- Related work analysis – analyze existing work and its results in the field of price prediction and machine learning
- Exploratory data analysis – conduct detailed description and analysis of available dataset.
- Missing values analysis – the aim is to examine dataset and find missing values which might harm our prediction model
- Data cleaning, preprocessing and preparation – to prepare data for model development and fitting
- Feature engineering – add open source data, transform the existing feature for better model performance
- Model development – develop accurate model by application of machine learning methods
- Model testing – test model and detect potential bugs
- Model tuning – in case of existing bugs, tune the model features for better performance
- Model evaluation – final stage of the research is evaluation of the developed model within the study
- Research documentation – document findings, developed model and code

1.4 Research Structure

The following section is dedicated for a reader to understand a big picture of the research on Machine Learning Application for Housing Price Prediction. It provides brief description of its structure, sections and outcomes. The study is divided into 5 main chapters: Introduction, Background, Methods for Prediction Model Development, Results and Conclusion.

The first section is introducing a scope of the research, focusing to understand underlying factors and requirements needed to successfully achieve set objectives. The chapter gives an outlook about field of study, research problem and its questions, methods and expected results of the thesis.

In the next section we pay attention to understand broader scope of the research problem. We look on the existing scientific as well as professional works, which might have common or overlapping project goals, ideas or outcomes. Moreover, the chapter provides fundamental knowledge to understand problem background, individual process steps as well as planned milestones and results. Not to stop only there, the section contains also concepts applied in the study.

The third chapter aims to describe methods, which are applied to achieve research results. The main content is focused on the process of prediction model development, where the reader might find information on the individual research steps, implementation and outcomes expectations.

The most important part of the research is focused to provide accomplished results. Starting with the description of the data, detailed exploratory analysis to comprehend underlying relationships and connections between specific potential price predictors. The chapter provides detailed documentation of the research, achievements as well as implementation of the applied methods. We analyze all aspects of the price prediction based on selected data and the journey to final outcomes. Conclusion chapter is aiming to give a constructive evaluation on applied methods and outcomes of the research. The chapter also documents possible further work of the research.

2 Background

2.1 Related Work

An ability to forecast future outcomes represents strong motivation for many professionals to conduct a research in this field. The advantages of data driven predictions are obvious; improved planning, personalized targeting and better decision making are few to be mentioned. There were conducted several studies, aiming to understand relationship between features of products and their correlation to price. In the following section we analysed works of researchers as well as business-oriented professionals, who were aiming to understand different aspects of price. The analysis is based on top-down approach, starting with broad researches and price prediction applications, to very specific ones.

2.1.1 Scientific Research

Housing price prediction is also interesting topic from scientific point of view. There were several researches conducted with the aim to predict prices of houses based on many different features. Most of the worked with publicly available data of land registry houses, based on few features by applying different regression methods [9]. What such researches are lacking is feature engineering, where it is important to work with most appropriate variables and try to include additional ones from publicly available data, which might also enhance performance of the models. Moreover, the studies of previous type are oriented on housing value prediction, not on the rent prediction.

There are very few researches and articles, which are trying to understand rent of housing and develop predictions. Most of them are older and based on hedonic pricing, based on mathematical modelling, functions and with focus on econometrics point of view [10]. Speaking about application of machine learning methods on price prediction of Airbnb listings, there are almost no comprehensive studies, helping understand price dependency on other accommodation variables. Airbnb was never providing data or any access to wider community. Recent integration of Airbnb API is helping users with more listings to better administrate their account. Airbnb was always cautions and for extraction of specific data it is necessary to use web scrapping methods. There are several projects which made these data accessible for research community, namely Tom Slee's project 'Airbnb Data Collection' [11] or 'Inside Airbnb Project' by Murray Cox [12].

2.1.2 Housing Service Providers

When it comes to housing price prediction, there are many companies and real estate service providers, who are trying to get ahead of competition with number and quality of offered services. The goal is clear: maximize profits and minimize costs. To keep a competitive advantage, it requires efforts in researching the market and understanding upcoming trends. Thanks to availability of data, companies are able to understand relations of many housing and renting features.

To start with biggest real estate service providers on U.S. market, there are several leading companies and association. First of them and one of the most significant is Zillow. Zillow is an online platform, based on data driven approach. The company has many successful projects, leveraging power of data. The ways how company does it are many. When it comes to forecasting and price prediction, Zillow is very active in providing wide range of insights on real estate market. Starting with basic statistics to advanced computations based on open data, Zillow provides variety of market insights for public [13].

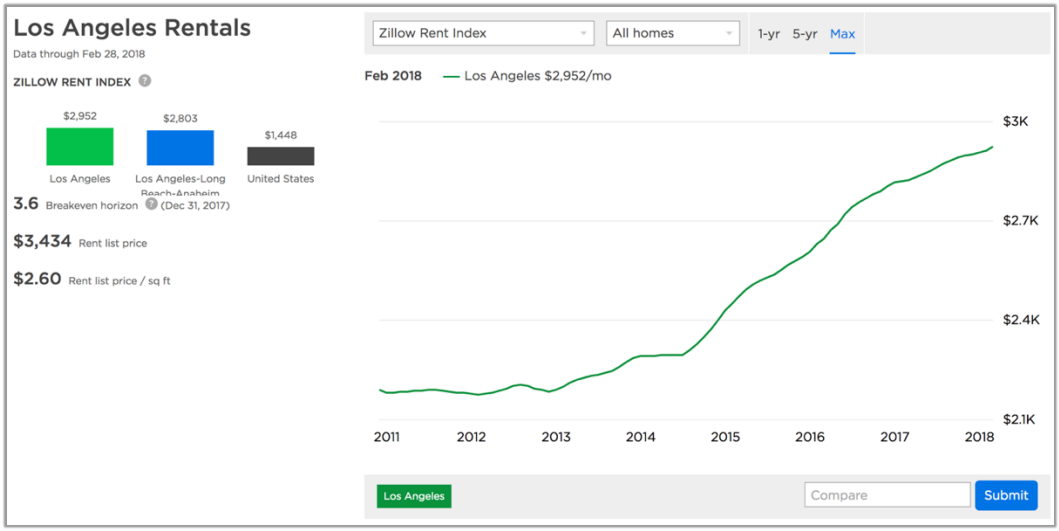


Figure 1 Example of Zillow's Rental Statistics in Los Angeles

The company established separate group of professional, called Zillow Research. By providing public with open and accurate market data, they help people better understand real estate market. Company's recent project, called Zestimate, is developed with the aim to let customers better understand price of their real estate in desired location and based on specific type of living. The functionality of our interest is Rent Zestimate, the service which predicts monthly rent. The aim of the application is to provide homeowner and landlords, how much is their property worth and renters about the price if it is about to be right. The Rent Zestimate evaluates more factors in rent prediction like: square footage, air conditioning, comparable rental properties in the neighbourhood [14].

Figure 2 Figure 4 Example of Rent Zestimate by Zillow

Rent Zestimate has an informative character. Company claims that the model is based on their best estimates, but price prediction doesn't have to reflect reality. The algorithm takes into consideration also publicly available data [15]. Although the service is great, it is not applied on Airbnb listings, with much different situation, type of listings and aimed for different target group, considering long term rental.

2.1.3 Airbnb Smart Pricing

Thanks to the quantity of information the Airbnb has, they are able to create rough estimation of prices, especially for hosts. The question is how accurate is company's price estimation towards hosts? Following section will be focus on the understanding the way how Airbnb estimation works in real life, for whom it is developed and where might be potential problem of algorithm. The hosts, willing to rent their accommodation has to face from the very beginning a struggle of how effectively set up pricing strategy, so they have booked available dates but don't underestimate the price and end up in minus in the end. And it is also a problem for company itself, as this step might discourage potential accommodations to be listed on the platform. The most intuitive way how a host might estimate price is to think in the dimensions of locality, especially, neighbourhood, quality of accommodation or quantity (number of beds, rooms, etc.). One of the possible solutions is to search for a similar rented accommodation, in the same area and estimate the price for individual's needs. But this is not convenient way how to convince potential customer to use the platform.

Airbnb therefore came up with a solution, which should support hosts in price estimation. What we found greatly confusing is a price recommendation from every side. Although we generally agree with an advanced price recommendation system, it might bring strong misunderstanding of which price is right for the proposed listing. When a user would like to

start with becoming a host, he or she immediately sees how much can earn in the city of property:

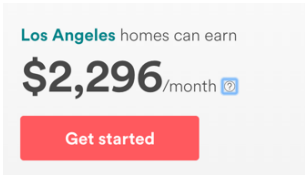


Figure 3 Example of Price Estimation by Airbnb

Based on deeper functionality research we understood that shown income is only in case when individual offers a whole place for minimum 6 persons, which might be very confusing for hosts with different rental specifications.

When we proceed to next steps, estimated price is changing based on parameters setting. More rooms, more accommodates, more beds mean higher earnings. But when we get to the last steps before uploading our offer online, we get detailed information about possible price setting. There are several prices, which we can choose from. Past year, low season, high season and the tip price.

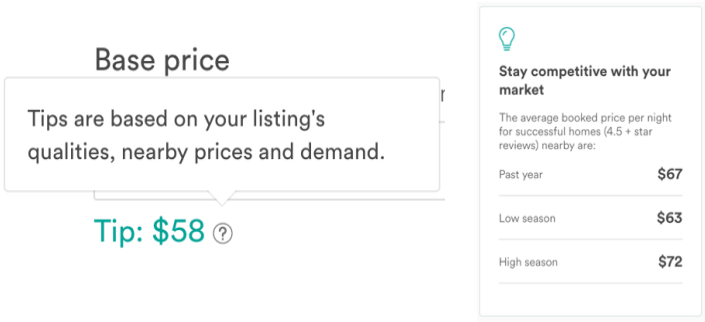


Figure 4 Example of Price Estimation by Airbnb

Considering seasonality within the price estimation is very helpful thing as the demand for accommodation in off-season might be lower. What would be interesting to know is if the algorithm is developed to incorporate extraordinary happenings, like concerts, sport events and others.

What we found especially interesting is that feedback for functionality itself is not always promising and satisfying. There are many complains, stating that price estimation is low and doesn't reflect reality. The Airbnb smart pricing consistently sets prices lower than optimal. The functionality is trying to maximize profits for the company, rent as much listing as possible and puts hosts on the last place. The goal of most hosts is not to be fully book but to maximize profits, but smart pricing seems to work oppositely.

From guest's point of view, there is no information about correct or recommended price in terms of available rentals. There is definitely a need for a functionality, which would recommend renters to estimate accurate price of listing like in previous example of Zillow's Zestimate.

2.2 Los Angeles Real Estate Market

Los Angeles or city of angels is located in southern part of United States, in sunny California. It is recognized by many significant attributes and is known as city of dreams. Over the years, it has become second largest city in USA. The city is one of the strongest economy driving cities with strong orientation on entertainment industry, culture, tourism and is well known by its huge number of museums, many attractions and famous events during the year.

2.2.1 City Characteristics

From the population point of view, it is second most occupied city after New York. Only city of Los Angeles has more than 4 millions of residents and when we talk about greater LA area, the population number goes beyond 10 millions. These numbers tell us about the size and diversity of the city. The population consists out of Hispanics, North Americans, African Americans, Native Americans, Asians and many European nationalities, with over 200 identified languages [16].

When we are talking about geography city of Los Angeles lies over a wide and diverse zone. It is situated next to the Pacific Ocean with more than 100 kilometres of coastal area, which makes a great condition for tourism and is surrounded by mountains and dessert. LA is consisting out of more than 85 smaller cities, famous by their luxurious housing areas and desired places for living. The city is famous by its neighbourhoods like Malibu, Hollywood, Beverly Hills, Chinatown, Koreatown, Venice, Pasadena, Anaheim or Little Tokyo [17].

The city is also very popular due to its amazing warm and sunny climate all over the year. Many tourists are heading there to get the required portion of sun rays and spend nice days near sea side. In general, we can say that LA is in high demand throughout a whole year. It is also famous by its incredible events. Starting with January, there is famous Martin Luther King Celebration and parade. February brings Chinese New Year, especially in Chinatown. In March we have Academy and Grammy Awards and popular Marathon. Very popular is also May with Cinco de Mayo. There are many events, which might represent a good signal for increase in demand for short term accommodations and rentals [17].

All these factors make LA a great place for life, holidays and relax. We are convinced that LA is city in demand and price of real estates will depend on all what we discussed previously. We believe that also these factors, directly or indirectly, are affecting prices of housing in this region.

2.2.2 Housing

Buying and renting a property in Los Angeles has never been more expensive as in the last months. The customers are facing strong increase in prices and no stopping growth in demand for accommodation in the city. This fact serves as confirmation that LA is one of the leading cities in United States, when it comes to housing industry. Investing into a property in LA was never more attractive as it is now. The factors, which are contributing the most are all, which we mentioned before. Economy, weather conditions, entertainment, population and tourism are few to name.

When it comes to prices of housing in Los Angeles, based on report by Core Logic, the median of real estate sale crossed $585,000, with over 6500 sales of homes, single family houses and condos in March 2018, which is increase of 6.6% in comparison with March, 2017 [18].

As we discussed before, trend in renting of real estate properties is also on the rise and Los Angeles is not an exception. With an increasing homes prices, many residents of the city are not able to afford to simply buy a house or flat. Based on the rent report of Los Angeles, prices for rents didn't move much in comparison with previous year. Considering past month, rent prices for one-bedroom apartment were in median at level of $1,360 and $1,740 for two-bedroom accommodation [19]. I

In conclusion, prices in LA are overpriced due to many reasons and keep rising from month to month. For regular people buying a property might seem unreal and they think twice if to rent or obtain new home. On the other side it is a seductive opportunity for investors or home owner to make extra income and offer a real estate to the hungry market. From the investor's point of view, Los Angeles represents a city of diverse investment opportunities. From traditional investing into real estates to Airbnb renting, the city is in great conditions.

2.2.3 Airbnb in Los Angeles

Los Angeles is a great place for home owners to offer their place and make something extra to cover mortgages and other costs of living. As the city is a frequent touristic stop, people who are coming to USA are often using services of Airbnb. Talking about statistics of Airbnb rentals in Los Angeles, in previous year of 2017 Airbnb hosts brought more than $1.4 billion,

with average host annual income of $9,100. These number are talking about great success of Airbnb. Services of Airbnb were used by more than million guests and impact are feeling local services as well [20].

2.3 Machine Learning

2.3.1 Machine Learning Outset

In the era of start-ups and emergence of new technologies, machine learning is currently in the center of research and development of the most leading enterprises. As a branch of artificial intelligence, methodology has been evolving over time with an objective to automate calculations, activities and processes based on knowledge extracted from its environment. Machine learning shares the fundamentals with mathematics, statistics and computer science.

From the historical point of view there are several events, which are closely connected with early application of machine learning methods. One of the early major events for this area of research is related to invention of first manually operated computer in 1940. The aim to build such system was to imitate human thinking and learning [21]. Later, in 1950, Alan Turing developed a test, which should estimate the learning capabilities of such machine. The test was designed to determine whether a computer is able to communicate, without being distinguished from another human. Another step forward in the hunt of advancements was when Arthur Samuel from IBM developed in 1952 a checkers game program, which challenged skills of many human players [22].

There were many other milestones, important for evolvement of machine learning and its wide adaptation, like the first designs of artificial neural network by Frank Rosenblatt in 1958, but the most significant advancements were accomplished in the last two decades. With the growing quantity of recorded data, their variety, increased performance and computation power of available technologies, application of statistical methods started to gain on popularity. From theoretical level of research, machine learning moved towards data-driven approaches.
Researchers started to develop solutions for analysis based on large-scale data, also called big data. Extending the calculation possibilities and easy accessibility and availability of data convinced companies to leverage this fact and include machine learning methods into their agenda. Creating new services for clients, improving processes, as well as products, is bringing to enterprises significant advantage in from of their competition. The field of research is still quite young but is getting more and more into the attention of big as well as smaller companies. The idea to predict outcomes of events is tempting for everyone. The

observations based on Google Trends shows an eminent growth in popularity of term “machine learning”:

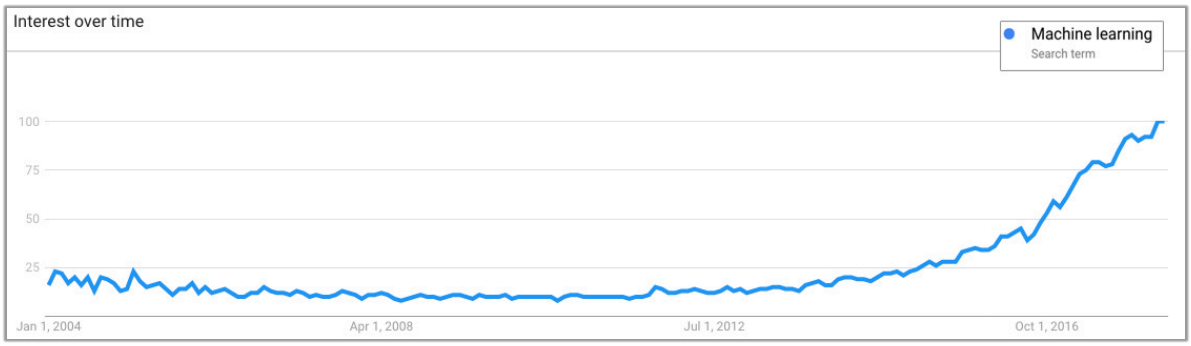


Figure 5 Google Trends of Machine Learning

The early adopters or pioneers in this field were technological giants. For instance, IBM's Watson computer with a wide range of use cases. From cognitive healthcare to support medical professionals diagnose diseases more accurately, through application in education, for holistic understanding of learning styles and personalization of the student's learning path to business messaging and many more [23]. Success implementation of machine learning in the biggest search engine provider, Google, and its Google Brain project for deep neural network research. Google Brain provided many interesting outcomes like object recognition, sorting quantum of images or language processing and speech recognition of YouTube videos analytics [24].

Facebook is also implementing sets of machine learning methods for face recognition with DeepFace project. Another utilization is to help user to discover new content, rank the feeds and ads relevant for them and let them free of unwanted spam. Also, well-known is computer vision to help blind people understand visual content by reading them recognized objects and comments [25]. Part of machine learning are also recommendation systems, most famous from companies like Amazon, Netflix or Spotify. Based on better understanding of customer preferences, the systems are able to classify needs and personalize individual services and enrich customer experience.

Potential use cases of machine learning might be in every sphere of interest. There are still underutilized industries where the appropriate application could represent a significant difference for companies. Insurance, healthcare, automotive industry, financial services, production, retail and many more are the best examples. The aim of every endeavour should be social good. One of such is also price prediction and there are not many services, helping individuals to better understand the price estimation. On one side supply or sellers, not to underestimate price in current conditions as well as on side of demand or buyers, not to overpay for offered services.

2.3.2 Machine Learning Definition

Machine learning is a field of study on the edge between statistics, mathematics and computer science and the term itself refers to “the automated detection of meaningful patterns in data” [26]. Fraud detection, teaching computer how to play chess, spam detection, image and voice recognition, prediction of outcomes or self-driven cars are few areas how we can apply machine learning. It is and always will be associated with computer. Simple difference between human and computer is that human learns based on mistakes and previous experience, but computer need to be programmed what to do. With machine learning, computers are also able to learn from past “experience” or in this case learn from data. We can describe machine learning as teaching computers to learn how to perform tasks from available data.

The field of study works mainly with statistical methods and algorithms and aims to develop models, which would be able to predict future or classify objects. Following illustration shows categorization of machine learning:

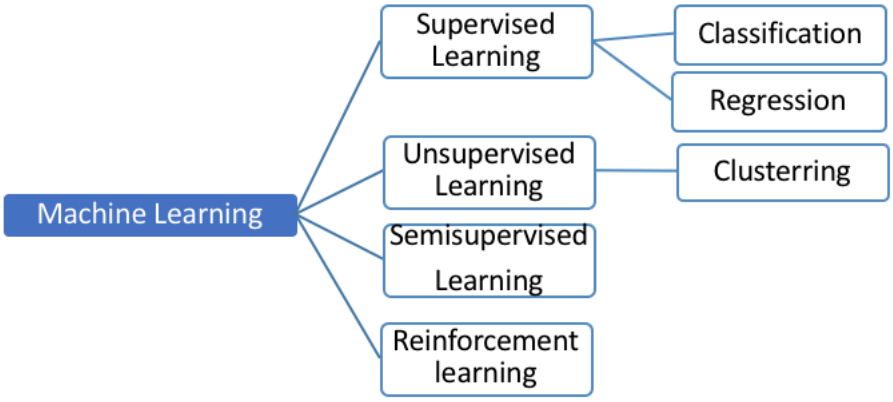


Figure 6 Machine Learning Categorization

Machine learning problems might be solved in different ways and by different algorithms. From high level perspective, we can distinguish two main algorithm categories: supervised learning and unsupervised learning. In supervised learning, we have knowledge how the results should look like. In other words, the objective of supervised methods is to develop a model, which would best fit expected results.

Unsupervised learning on the other hand, can identify patterns and similarities in dataset without a supervision of human [27]. It is able to uncover hidden structures in unlabelled data. The main objective is to discover unexpected outcomes from the observations and methodologies are more subjective in comparison with supervised learning, where the aim

is to develop predictions based on previous experience. As a base are considered clustering techniques for creating some similar groups, such as: grouping cancer patients based on gene measurements or movie clustering based on ratings. The importance is hidden in its abilities to find unexpected relationships and from unlabelled dataset create grouped structures. Unsupervised learning might be also used for data pre-processing or in exploratory data analysis. Great example of this use case is principal components analysis, or PCA, which derives variables for supervised machine learning [28]. Deciding between supervised or unsupervised machine learning algorithms is usually determined by characteristics of the structure and volume of observations and the use case of the problem. unsupervised learning is related to AI, or artificial intelligence – where machine learns to comprehend complex patterns, problems or answers for tasks without a human supervision. Still, supervised learning is more frequent and used for problem solving.

On the boarders of unsupervised and supervised learning there is semi-supervised learning, which uses both labelled and unlabelled data. The common use cases are for speech recognition or classification of webpages. Advantage is that with the correct application of semi supervised learning we can leverage from big volumes of unlabelled data, together with supervised techniques for smaller amount of labelled data [29].

There is also reinforcement learning, which is a type of machine learning, where machines or so called 'software agents' try to figure out, how to optimize their behaviour based on the system of punishments for undesirable actions and rewards for desirable actions. The most significant sign of reinforcement learning is trial and error. The agent is not instructed what steps need to do, but on the other hand has to find the steps which lead to biggest reward. A good example is vacuum robot, who needs to decide if to continue with finding more dirt or return back to charging station. Based on previous experience, how easy or difficult it was to return back to charging station, it makes decision if to continue or go to recharge [30].

Application of machine learning is greatly wide and might change every field. There are many different algorithms, which might be applied for variety of all kinds of situations and everything depends on objectives and available data. In our research we will utilize a power of supervised learning and its algorithms for predictions.

2.3.3 Supervised Learning

Supervised learning is one of the most used set of algorithms for machine learning and is based on two ways of problem solving: classification and regression. The method enables programs to classify data based on passed knowledge. Supervised learning takes inputs and outputs and based on new inputs predicts its new outputs. It enables the data to be classified based on chosen features and derive predictions out of it. There are many tasks

which might be solved by application of algorithms based on supervised learning. The main objective is to develop a model, which creates predictions based on experience out of the present observations [31]. Following figure explains how supervised machine learning provide predictions:

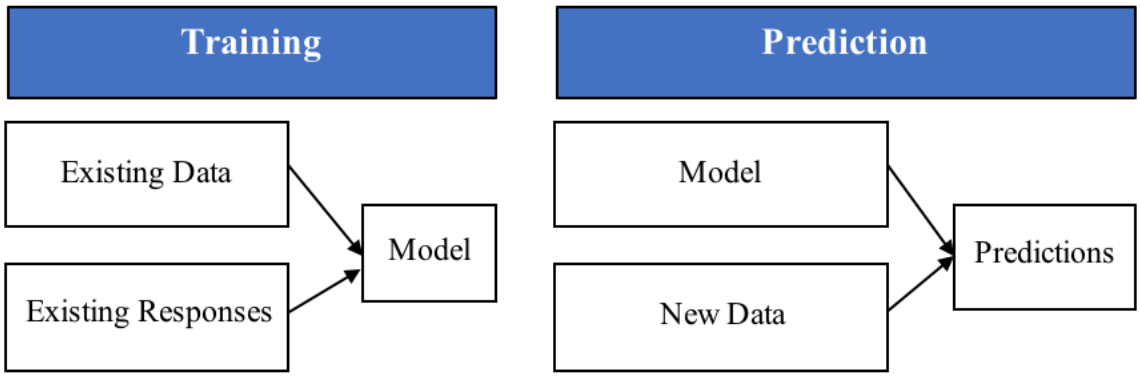


Figure 7 Supervised Learning Model

In supervised learning, we begin with loading existing data, which has available trainings features and training targets. Model will train and learn existing relationships between observations and their target variable. Predictions are then conducted based on trained model and new data inputs [32]. In the training process supervised learning fine tune parameters by comparing current output with expected output. As a feedback during the training process, the model works with error measure. It represents the divergence between the output from training model and the desired output. This error feedback is in supervised machine learning described by mean square error function.

In real life, models are learning on training dataset, which contains labelled examples and are tested on unlabelled datasets. It is important to tune the model on training dataset, so the performance on the test set is as high as possible. To define supervised machine learning more mathematically, the training set contains m ordered couples – x_1, y_1 and x_2, y_2 – where x is a variable based on which we are trying to predict y target variable. The indexes are representing type of dataset – 1 stands for train and 2 for test dataset. As mentioned above, the objective is to make rational guesses about the labels or target variables in test data by gaining insights from training data [33].As already mentioned before, most of the models and predicting applications are taking as a base methods and algorithms from supervised machine learning. Based on the dataset as well as task which the model has to do we distinguish two different approaches to problem solving in terms of supervised machine learning: regression tasks and classification tasks.

Regression tasks are related to predicting or estimating certain outcomes, where the value has continuous character. Typical regression problems are trying to answer questions of following type: What will be the value of a real estate property next month, how many new

customers will our shop acquire this year? To find the right solution for the mentioned tasks we need to find the best fitting regression model. In case of regression, the most common way how to measure performance of the prediction model is mean square error. Following are typical algorithms of regression, used in regression-oriented tasks [34]:

- Linear regression
- Ordinary least square regression
- Lasso regression
- Multivariate regression algorithm
- Generalized linear model
- Ridge regression
- Regularized regression
- Support vector regression
- Ensemble methods
- Etc.

Another subgroup of supervised machine learning is trying to solve classification problems, where output of the estimation or prediction is a classified category based on specific features or similarities. In the classification tasks, models are predicting discrete values, where it takes input and estimates to which class, out of existing ones, will it be assigned to [35]. The fact that the learning is based on the existing classes or categories, classification is considered as supervised learning. But the discrete way of classification is not so straight forward in real life. There are usually many observations which doesn't fit to any of predicting groups, but rather in between of two or three groups. Very similar to classification is categorization, which belongs to unsupervised learning. The difference is, that in first case we have specified groups and in the second one, groups are created on the way, by an algorithm itself. Examples of questions to be answered by classification are: Will the stock prices rise next year? Is an email a spam? Is it red or yellow? Will the price of real estate grow or not? As it is obvious from the examples questions, in classification, model is aiming to group data together based on particular expectations [36]. Following algorithms or models are typical for classification tasks:

- Linear classifiers
- Naive bayes classifier
- Support vector machines
- Discriminant analysis
- Nearest neighbors
- Decision trees
- Etc.

Difference between regression and classification that classification is oriented on the estimation of discrete values, but on the other hand, regression is solving tasks with continuous quantity. Following illustration shows difference:

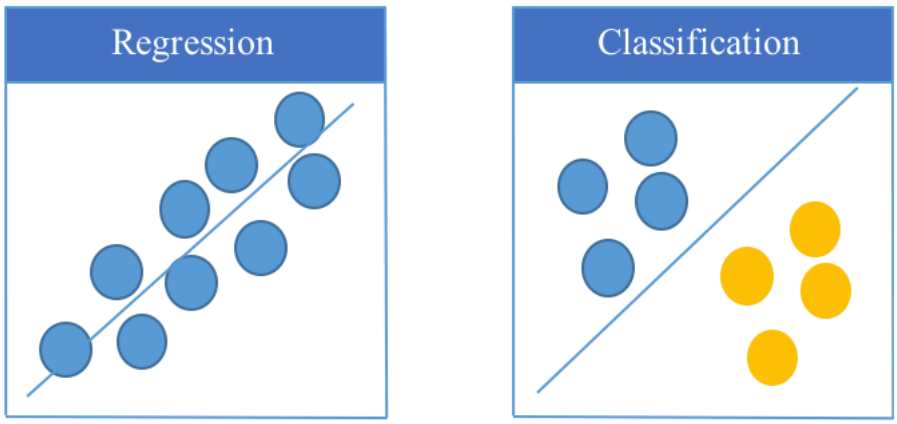


Figure 8 Regression vs. Classification

Moreover, the difference is in the way how we evaluate models – in classification we are considering performance of predictions but in regression-based model we look on root mean squared error. It is important to note that some algorithms are suited for both, classification and regression tasks, such as decision trees or neural networks [37].

2.3.4 Models for Housing Price Prediction

Housing price prediction is a task which deserves attention of many researchers due its importance and necessity for both parties of sales: customers and sellers. There were several researches done in terms of Housing price prediction, but few with the focus on rent prices, especially when it comes to such a frequent service as Airbnb. People renting a property, either on the hosting side or renting side, are drivers to understand the task better. When we talk about listing on Airbnb, there are many approaches which might be utilized for model price prediction development. It depends on the nature of the data, its size and quality, which way should the professionals approach the task. Except data, there are many factors, how to decide which algorithm to choose from. Computational time, power or hardware which we have available are also playing a role.

Fundamental base of how to select a model is to classify a kind of task, which we want to do. Starting with the types of supervised and unsupervised learning. As we mentioned before, realizing the nature of data is crucial step, in case we have labeled data, we go with supervised machine learning. In the case of Airbnb, data are labeled, and we will closer examine the algorithms relevant for our modelling.

Furthermore, supervised learning is trying to cope with two common problems of classification and regression. We can solve the task with both ways, in regard to Airbnb listings:

- As a classification task: to develop price groups of clusters, with specific rages, where we would try to classify our predicting price. This would allow individual parties, interested about price of listing, to understand in better and decide if the price is low or high. A challenge might represent a development of such clusters.
- As a regression task: As a price has characteristics of continuous value, we can intuitively classify a price prediction of Airbnb listing as regression task. We are aiming to answer what will be a value of the Airbnb listing with specific attributes?

In our research we will be solving Airbnb listing price prediction with second approach, where we will utilize power of regression methods. There are important approaches and techniques which has to be applied within a process of prediction model development which we need to fulfil in supervised regression tasks. We will focus on those in the next chapters. In the following section we will introduce algorithms and models which are relevant for our regression task ad which will be tested. The chosen algorithms represent best potential to make our model successful. We will describe following algorithms: Linear, Ridge, Lasso, Elastic Net, Bayes Ridge, Random Forest Regression.

2.3.5 Ordinary Least Squares Regression

Standard linear regression model or also known as ordinary least squares model. The aim objective of this model is to understand how are features, containing values, interacting with labelled values or results, dependent variable. To apply this kind of model on our data represents to predict the values of the features' coefficients, which are part of the model.

The Ordinary Least Squares or shortly OLS is widely used when we have more than 1 input for prediction of outcomes. The aim of the procedure is to minimize the sum of squared errors from linear line. In other words, we can say that the algorithm is computing sum of squared errors and fitting best possible line, with the lowest sum of squared errors within data, provided to create a model [38]. Depending on the quantity of features applied for prediction, OLS might be simple or multiple. Following formula describes Ordinary Least Squares regression:

$$Y = \beta_0 + \Sigma_{j=1..p} \beta_j X_j + \varepsilon$$

To understand equation, the result is our target predicting value, β_0 is representing intercept, β_j is independent variable, X is selected feature and ε stands for error term [39].

When we are referring to advantages of the OLS we definitely can say that it is simple to comprehend. Understanding of linear regression is straightforward. Moreover, computations are fast, and the method is powerful with simple datasets. On the other side, it is not effective when it comes to more complex samples and relationships within individual variables. The higher number of predictors the less effective is OLS. Therefore, there are some modifications of the OLS, which are enhancing performance of linear models. We are talking about penalized regressions, known as regularization techniques, which are perfect for multivariate datasets [40].

2.3.6 Ridge Regression

All the following algorithms or regression models are based on regularization techniques, which are aiming to decrease importance of features with small impact on predicting variable. Regularization plays important role in the process of avoiding overfitting. When applying regression methods, we are commonly dealing with loss function, where regularization is shrinking the sum of square errors by choosing right coefficients [41].

Ridge Regression is utilizing so call L2 norm. The decrease of the coefficients is accomplished by penalizing the regression model with the sum of the squared coefficients [42]. It is good to note that instead of getting rid of all less important predictors, we are keeping them as a part of model but will smaller coefficients. Following equation describes Ridge Regression:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right) + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2$$

The equation above is explaining the ridge regression, where RSS is adapted by including decreasing attribute. This way the coefficients are predicted with minimization of loss function [43]. Lambda sign is representing a tuning parameter, which gives penalization to the prediction model. In case of Ridge regression, we need to standardize selected features or use feature scaling.

2.3.7 Lasso Regression

Lasso Regression is very similar method to Ridge Regression. The difference is in the way, how we are penalizing coefficients. Ridge Regression has also some drawbacks and one of

them is its interpretability in case of higher number of variables. The limitation comes with the way how it penalizes coefficients. The algorithm will not drop any of features, rather keep all of them, even when coefficients are heading towards 0. This limitation has been addressed by quite recent method of Lasso Regression. The coefficients of Lasso are minimizing quantity [43]. It conducts L1 norm, which is utilizing absolute value of the coefficient's magnitude. Following is the formula of Lasso Regression:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right) + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$$

As we can see, the way how lasso is working is almost identical with ridge, despite the fact that predictors are penalized towards zero and some of them will end up equal to 0. It means that those will be eliminated, and the prediction model will be reduced of those features. For the model interpretation it is simpler to focus on less features then keep also the ones without almost any impact.

2.3.8 Elastic Net Regression

Elastic Net is a method of regression where we are implementing the power of both previously discussed models. The aim is to decrease specific coefficients close to zero as well as set some of them to zero [44]. Elastic net is applying both norms of regularization, L1 and L2. In case we have more predictors in our dataset with stronger relationship, Elastic Net Regression will create a subset of them, which are correlating and contributing to the prediction model performance. In case one of the features, which are grouped in the subset, is highly correlating with dependent variable, we will develop a model together with the rest of the features of the subset. Reason is clear: as we would reduce subset just by 1 variable it might end up with loss of information and unsatisfactory results of model [45]. The penalization of Elastic Net is then as follows:

$$P_\partial = \sum_{i=1}^{n}\left(\frac{1}{2}(1-\partial)\beta_j^2 + \partial|\beta_j|\right)$$

This is how the penalization of Elastic Net Regression working, taking both into consideration: L1 and L2 norms [46].

2.3.9 Extreme Gradient Boosting

Extreme Gradient boosting or shortly XGBoost has recently become one of the most popular algorithms within machine learning professionals. Extreme boosting is powerful algorithm based on gradient boosted decision trees for faster and better performance.

The advantage of the XGBoost is that it used for both types of supervised machine learning with x_1 training set to find dependent variable y_1: classification as well as regression. It utilizes more regularized model formularization to better deal with overfitting in aim to increase performance of the model [47].

2.3.10 Bayes Ridge Regression

Bayesian approach to regression differs in the way how regularization is done. The parameters are tuned while prediction procedure is conducted. In comparison with Ridge Regression, where the coefficients are gathered by L2 norm, Bayesian Ridge Regression shrinks the coefficients more heavily [48]. In other words, parameters are predicted while the modelling by maximizing the marginal log likelihood [49].

The main objective of Bayesian Linear Regression is not to discover the only one most desired value of the model parameters, but instead to find subsequent distribution for that parameters. As the crucial coefficient becomes solid, the performance of the model increase. Bayesian approach to the problem solving might be difficult and beyond control but on the other side might bring expected results and better performance.

2.3.11 Support Vector Regressor

Support Vector Machine is based on supervised machine learning principles and in able to tackle with both types of tasks: classification as well as regression. Even though it is mostly applied for classification, it is also used by many for regression purposes in the form of Support Vector Regressor or shortly SVR.

The main goal of the algorithm stays the same in both cases: to minimize error by finding the hyperplane that maximizes the margin between two classes [50]. SVR is classified as a nonparametric method due to its dependability on kernel usage. The aim is to identify a function f(x) which differs from y_n by an amount not bigger than ε for every observation x, which staying as horizontal as possible [51].

The advantage of the SVR is its prediction performance. The algorithm is able to give very good estimates. It also tends to avoid overfitting. On the other side of the coin is its computational price. The calculations are very demanding and require much more resources than simple linear regression. The algorithm might run slow even on slightly larger datasets [52].

3 Methods for Prediction Model Development

Before the implementation of any price prediction model into production, there is a set of methods, which has to be conducted to get the best model possible. There are many concerns, which is necessary to keep in mind upfront. First and the most important part of any machine learning project is data. Implementing prediction models into any operation might be intimidating and requiring process, but it is a way how to discover knowledge and provide better services. The performance of the prediction model depends on following:

1. Quality of Data: to get the prediction model working with the high performance, data has to be of certain quality. There are many techniques and methods which are dealing with skewed data, missing data or different format of data. There are cases when data needs to be transformed or changed in some way. Most of the data comes in raw format, needs to be examined, explored understood and pre-processed for further use. It is also important to let machine learn or train on one data split and test on the other one. The goal is to keep unbiased process for better efficiency [53]

2. Feature Selection: even we have data in the best possible quality, not all columns, variables or features has to be related to each other and might play minor role when it comes to model development. Crucial for machine learning process is to set the goal which we want to achieve by predictive data modelling and then discover the most relevant elements. In this case we are talking about features influencing target variable, in our case price or rent. Feature selection is the technique of choosing a subset of attributes, to be incorporated in prediction model. Most of the times, a right feature selection is a base for better and powerful model.

3. Model Selection: last but not least in the quest for the most accurate predictions is selection of algorithms. There are many machine learning tasks which might be approached in different way. Performance, speed, quantity of data or type of the problem are few factors to name when it comes to model selection [54].

For all mentioned points, there are methods and techniques which we need to consider, examine and based on consequences, apply to build our data model. Starting from data exploration, cleansing, pre-processing and model selection we will examine all the available techniques to develop our prediction model based on machine learning methods.

3.1 Exploratory Data Analysis

Exploratory data analysis, or shortly EDA, is, as the name already describes, an analysis aimed to view on the data from different angles. It is an approach how the data analysis should be conducted in the right manner. It utilizes several different techniques and support understanding of data. EDA is mostly visual analysis of many aspects of data – from basic statistics to more advanced relationship bonds. Even though the form of EDA is mostly graphical, it also operates with many quantitative methods. Revealing structures of data, finding new patterns, identifying outliers or gaining unexpected insights are the main objectives of the analysis. Moreover, thanks to EDA we are able to get strong signals about which models would be the best fitting for our prediction objectives. Because the analysis is mainly visual, it employs plotting and graphs approaches like: histograms, scatter plots, box plots, pie charts or dot plots [55].

The advantage of EDA is in its ability to present and utilize all the available data with no loss of information. The objectives, which should be met after exploratory data analysis are.

- Understanding of data
- Findings of underlying structures in data
- A list of crucial variables
- Identified outliers
- Elemental hypotheses tests
- Suggestion of optimal model
- Idea about parameters of features

After the correct application of EDA, research should obtain awareness about dataset, or what it contains and what doesn't [56].

3.2 Data Preprocessing

Data plays crucial role in the process of machine learning. It is a first asset which is responsible for the further steps in the development of prediction model and if it is not selected in the right manner, anything else might be priceless. There are many attributes of data, which are required to be met for highest model performance. Format, chosen variables, data type – all these are playing crucial and important role when applying machine learning methods in any sphere of machine learning research [57].

One of the most time-consuming and demanding phase of predictive modelling is data pre-processing and preparation. There are several methods, which might be applied along the way of machine learning. After data selection, there are requirements, which we might

consider, to get data ready for prediction model. Data are pre-processed by many techniques, like splitting data into train and test sets, data cleaning, unsupervised pre-processing, with techniques of data addition or removing, data transformation, skewness reduction, removal of outliers, feature engineering, feature selection, feature extraction, handling with missing values and more. There are factors like type of task, data and other which need to be considered before applying above mentioned techniques [58].

3.3 Feature Engineering

First things first, understanding features is necessary for predictive modelling. As we have already discussed, machine learning models are learning from set of collected observation and based on them conclude prediction of the new input values. Predictions are based on available data and their attributes. Attributes have specific characteristics and they make them special. A feature, based on Bishop, is an 'individual measurable property or characteristic of a phenomenon being observed' [59]. It is an attribute that is convenient or appropriate for a task solution. The significance of features is so high, that even a model based on few correct features might have much better results than complex one with many weak features. A feature is crucial when it is highly correlated with the target or dependent variable. Here is the moment, when feature engineering comes into game.

Feature engineering is one of the most relevant methods for successful predictive model development and is responsible for encoding of individual features [60]. Also, Andrew Ng, one of the most visible persons in machine learning, said: "Coming up with features is difficult, time-consuming, requires expert knowledge. Applied machine learning is basically feature engineering."

Feature engineering represents a significant challenge and is strongly depending on human intuition to understand specific characteristics of data. It is closely related to data transformation and contains several techniques how adjust them. Following are the categories of feature engineering: adjustment of feature representation, engineering of interaction features, engineering of indicator features and external data integration [61].

Adjustment of feature representation is easy, but very powerful category of feature engineering. The format of data plays for machine learning a crucial role. It has to be taken into account that different format of variable might add greater value to prediction model. Examples are 'date' variables – instead of exact date it can be more convenient to represent a variable as a day of the week, month, etc.

Engineering of interaction features represents a way how to point out relationship or synergy between specific variables. Combining those variables might bring more light into the

darkness then if they would stay apart. Great example operations are summing variables, subtraction, multiplication or division.

Engineering of indicator features means to adjust or create a new variable to indicate something specific, what is clear for human understanding, but difficult for machine understanding. Abstracting fundamental information might point out something hidden. Examples might be to create an indicator feature, to flag special events of the year, like Christmas, sport events, spring break, etc. We can also think about indicator feature from more variables – to indicate that observation contains combination of specific values.

External data sources are always a good way how to possibly increase a scope of predictive modeling and can enhance the results of applied algorithms. A lot of machine learning tasks can leverage the power of open source data or other external data. Great examples are websites offering APIs. Another example is geocoding. If the dataset contains overlapping variables, it might be merged and bring new variables. Same applies to time series features. If we have information about date of booking, we might to increase our dataset with other data, for example related to season.

There are still many other ways how to engineer features. It is important to keep in mind that feature engineering represents a huge role when we talk about successful prediction model.

3.4 Feature Scaling

Feature scaling is one of the methods, which might be applied in the data pre-processing stage of predictive modelling. As most of the variables come in different formats, scales, units or ranges, feature scaling might play important role in machine learning process. It might vary a performance in case of some algorithms. The objective of feature scaling is to standardize a scale of features [62]. If a scale of selected feature is much larger or much smaller than the other variables, performance of the algorithm might be strongly affected by that feature [63]. To solve this task or imbalanced variables, we use feature scaling methods by bringing all features into same rage or scale.

One common way how to avoid an issue with scales of features is to utilize power of min-max scaling. The scaler rescales variables to a specific rage, usually between 0 and 1. Following formula shows how min-max scaling is conducted:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Another way how to scale features in required rage is by data standardization, also called Z-score normalization. The concept of standardization rescales features in the way that they get attributes of standard normal distribution with μ=0 and σ=1, where μ represents the mean and σ is the standard deviation [64]. Following equation shows how standardization or Z-score is conducted:

$$z = \frac{x - \mu}{\sigma}$$

The outcome of z-score calculation is that variables will obtain properties of standard normal distribution and values are replaced by their Z scores. Many algorithms used for machine learning even require variables centred around 0 with σ=1. Good examples of algorithms, requiring feature normalization are: K-nearest neighbours, logistic regression, SVMs, neural networks. Another instance, where normalization is required is principle component analysis or PCA, where we would like to know, which features maximize variance. We need to have features of same scales as PCA emphasize features with higher number scales [65].

The application of either one or another method depends on algorithm we are using. There are other ones, also used to normalize unit vector scaling or mean normalization. In our research we will utilize the power of feature scaling with Python open source library for machine learning called scikit-learn.

3.5 Feature Selection

Nowadays datasets are containing a lot of data, with many columns, representing attributes. It is not extraordinary to see even hundreds or thousands of features. This fact is responsible for multidimensionality [66]. Feature selection is a fundamental part of predictive modelling. To choose right dataset representatives means if the model will work correctly or not. The process of feature selection means to choose the most applicable ones for the task of prediction model needs to accomplish. The selection aims to reduce number of features without combining them. It is behaving as a filter method, excludes unnecessary attributes and keeps the one which are playing an important role for prediction. Less features means, less complex model, better performance and better comprehension. The model should only be based on features that contribute to the most accurate predictions. Result of feature selection is a subset of features, not containing irrelevant or unnecessary ones [67]. Reasons behind feature selection are following:

- More features represent complex interpretation of model
- More features mean more time for training
- Models with more features have higher chance of overfitting

The aim of feature selection is to avoid mentioned issues by reducing number of features for prediction model. Feature selection is also helpful to estimate influence of individual features on target variable. It consists of methods, which are helping to solve mentioned problems. There are 3 most common methods, how to select features:

- Filter
- Wrapper
- Embedded

Filter methods are examining an affiliation of target variable with the other features. The main objective of these methods is to calculate a score of a variable importance to target one. Based on the results, the feature is either further considered for prediction model or dropped from dataset. Common methods are information gain, Pearson's correlation, correlation coefficient scores, anova, linear discriminant analysis, chi square, f test or variance threshold. If needed, individual methods might be combined in order to get better performance. It is also crucial to consider, what type of data we have – continuous or categorical [68].

The other way how to select features is to use wrapper methods. The core of wrapper methods is to select a subset of features as a wrapper around the learning algorithm. The aim is to find the best subset of features and it is considered as a search task. A learning algorithm is trained on dataset with different subsets of features to find the one, with the best performance. Various solutions are tested, compared to one another and evaluated. [69]. This method might be computationally highly demanding. Following are examples of feature selection wrapper methods: forward selection, backward elimination, recursive feature elimination.

There are few differences between filter and wrapper methods. First of all, it is a way how an importance of features is measured. In comparison with filter method, where there is single one variable's importance measured towards dependent variable, in wrapper methods we look only on the best suitable feature subset as a whole. Filter methods are also les computationally expensive then wrapper methods, where the features are selected directly on the training prediction model. On the other hand, wrapper methods might be more reliable and usually resulting in the best feature subset for prediction model, which might avoid overfitting or under fitting [68].

Last but not least we have embedded methods for feature selection. These methods are combination of previous two approaches. Embedded methods select features during the learning process. In comparison with wrappers methods, embedded methods don't split learning process and feature selection. It means that the search for best performing feature subset is running during the model learning [70]. Typical embedded methods are

regularization algorithms like: lasso regression, ridge regression, regularized trees or elastic net.

3.6 Feature Extraction

Feature extraction is closely related to dimensionality reduction, where we aim to decrease number of dimensions of selected dataset in order to achieve better performance or more effective computational results of prediction model. Feature extraction is aiming to reduce quantity of needed assets or attributes, which characterize data. There are several reasons why to consider feature extraction as a required part of pre-processing: when a dataset contains too many variables, it is computationally expensive and demanding a lot of power, memory. Another case might be with overfitting of training set and as a result poor performance of prediction model. Unlike feature selection methods, feature extraction does not reduce dimensionality by dropping number of features but rather adjust the initial form of data [71]. Most of the times we would like to decrease number of features in order to increase performance, make the prediction model computationally not expensive but rather effective, with no loss of information. The question is “how to choose right features and not lose any information? There are several ways how to extract features. Most common, standalone methods are Principal Component Analysis or Linear Discriminant Analysis.

Principal component analysis or shortly PCA is a method utilized to find relationships in data, identify patterns, present strong similarities and differences in data and reduce dimensionality of large sets without loss of information [72]. The method is connected with multivariate analysis, where we are trying to find relationships between bigger number of features. PCA is often used as a great technique for data analysis, where large dimensional datasets might be compressed.

The goal of the analysis is to select smaller set of features with the highest correlation. The main part of PCA are principal components – result of transformation of correlated features into smaller subset of uncorrelated or orthogonal features. The aim is to discover linear relationships of principal components with highest variance. PCA is used in cases when we have unclear, complex and large datasets, with redundant observation and with hidden relationships between possibly correlated variables [73]. There are several reasons why to use PCA as a great method of feature extraction: it is easy to implement, supported both by Python and R and it comes in more alternation to solve unique cases. The other side of coin in case of PCA is that created principal components are not possible to be interpreted.

Linear Discriminant Analysis or also called LDA is widely utilize for reduction of dimensions in many machine learning prediction models. The aim is the same as for PCA or other feature extraction methods, to decrease number of variables, decrease requirements for more

computational resources and escape overfitting. LDA and PCA are comparable but LDA is also examining the separation instead variance. Both methods are coming from same linear category, but LDA belongs to supervised algorithms, which are considering labelled data. PCA is on the other side unsupervised oriented and doesn't take into account labels [74].

3.7 Missing Data Imputation

Based on previous sections we can say that preparation and pre-processing of data requires a lot of time and efforts. There are many steps which we need to be fulfilled in order to have prediction models with the best performance possible. As we already discussed, data might come in different formats, with large number of correlated and uncorrelated variables, different scales or rages or different data types. But what can we do with the data if there are missing values? Do we necessarily need to delete whole observations without complete values even if they represent strong potential to be used in prediction model? As dealing with missing values might be challenging and not always rewarding, in some cases it might help to avoid information loss and to improve overall results. In case of supervised predictive modelling the aim is to develop a model on labeled data. If there are missing values, it might represent a problem [75].

We are referring to missing data, when an observation is missing one or more values in the representing features. There are many reasons why data might have missing values, such as wrong measurements, manual data insertion failure, incomplete or wrong data entry, poor data collection, censored or not publish values, data files might be corrupted or involved parties refused to provide values. It is crucial to understand underlying reason behind missing data to deal with the task in the right manner. Data might be missing randomly or systematically [76]. Based on the reasons we distinguish types or mechanism of missing data:

- Data are missing completely at random
- Data are missing at random
- Data are not missing at random

Data missing completely at random or shortly MCAR is a mechanism of missing data where there is no connection or systematic reasoning behind missing values. In other words, it means that there is random subset of values which are missing and no tendency for other observation to have missing data as well. MCAR type of missing data says that relationship between missing values of one observation and another is not correlated and is completely random [77]. Data are missing at random or shortly MAR is a type, where missing values are related to the observation itself and can be predicted based on other information of observation. In case that observation has missing values, it can be imputed based on some other variables. Good example of such case is that some specific group of people, for

example women or poorer people are not willing to provide answer about some sensitive topic [78].

Common type of missing data is also data are missing not at random or shortly MNAR. It means that there is a tendency of relationship between missing values and observation itself. It means that the missing values are in direct correlation to the observed event, individual or record. Example might be a rich person who don't like to provide information about property value or person who in not willing to conduct an alcohol test because he or she drunk before measurement and etc. [79]. There are few ways how to deal with these types of missing values:

- Get missing values
- Drop or delete the observations with missing values
- Alter missing values by prediction based on other observations
- Predict missing values based on other attributes of the observation

When it comes to MCAR and MAR we can consider total removal of observations with missing values. When we face MNAR we need to consider different approach, conduct analysis of missing values and understand it better. Afterwards we can decide about next steps of removal as it doesn't have to necessarily mean better performance of model or imputation of missing values. Dealing with missing values might be tricky, even dangerous when it comes to imputation of data. Sometimes we need to rely on intuition and experience when we need to decide whether to delete or impute missing values. On the other side deleting observations with missing values might result in biased and ineffective prediction model. We also need to distinguish different approaches when deleting and imputing missing data.

Deletion might be listwise, pairwise or total variable dropping. Listwise deletion or row deletions is aiming to delete whole observations with one or more missing values. This method of deletion might be ineffective for model performance as in might cause biased predictions. Pairwise approach to deletion is based on minimization of information loss. Advantage of this method is that it supports data analysis and weakens biased results. On the other side, results of pairwise deletion might be different amount of records for each model prediction component. In the end we end up with complex model for interpretation. This fact is also responsible for overfitting or under fitting [80]. Last but not least is variables dropping, which comes as the last choice in case of bigger number of variable missing values. It is always think twice if it is necessary to delete the whole variable. We need to consider if a variable is important or not, if it plays crucial role or if it might support our prediction model. There are many discussion, when variable should be dropped, when not. It always depends on type of variable, correlation to dependent variable and of course amount of missing values. To start considering if we drop variable or not, there should be at

least 10-15% of missing values. There might be even extreme cases when variable is missing 50% of values. Most common recommendation of missing values variables removal is between 25-30%. It is always to keep information and not delete whole predictor, but less predictors means decreased computational requirements and complexity.

For the imputation of missing values in data, there are many methods which are utilized. Starting with common technique of mean, median and mode, which is very basic, simple, time saving but in the same time brings together with it many possible harms as it might change variance in data. This kind of computation might be only used on continuous or numeric data like age, or apartment rents, etc. Advantages of mentioned statistical methods are mostly visible on smaller datasets, where it can avoid information loss. Disadvantages are connected with variance and bias change [81].

Another way how to impute missing values is K Nearest Neighbors algorithm or KNN. It is widely utilized and is commonly used for continuous, discrete, ordinal as well as categorical missing data. To determine missing value with KNN, algorithm firstly picks k nearest neighbors and calculate their mean. For this technique it is necessary to choose number of nearest neighbors and distance metric [77]. The fundamental basics of KNN is that missing value might be estimated based on near values of other variables. There are parameters which have to be considered before applying KNN in regard to data type, number of neighbors, attribute distances, data normalization and so on [82].

Missing values of data might be also imputed by multiple imputation method. The way how multiple imputation method works is that it recovers missing data by set of possible and probable values. The values which were replaced are then analyzed as a normal dataset without missing values and outcomes of both analyses are then incorporated in predictive modelling [83].

There are other methods for missing values imputation like linear regression, which utilize other available variables in dataset. Features with missing values, which might have relationship with target variable are considered as dependent variables in iterative process, where features with complete data are used for prediction of variable's missing values with regression approach. In this sense there might be also different algorithms used for missing value prediction in similar manner [81]. When it comes to Time-Series data, there are also methods which are dealing with this task. Last Observation Carried Forward or Next Observation Carried Backward, Linear Interpolation or Seasonal Adjustments are few to name.

To sum up topic of missing values, it is always better to keep data and not loose information. It is necessary to consider if the feature is necessary for our prediction model or it would be

better to drop it. In the end, less predictors means less computational requirements, less complex model and better interpretability.

3.8 Outliers Identification

Outliers are values in data, which based on their characteristics are extreme and deviate from the rest of records and is considered as outcast of dataset. The reasons behind outlying data are entry errors caused by humans, measurement anomalies and errors, processing mistakes, novelty in data also known as true outlier or others. Outlier also comes in different types. Based on the dimensionality we recognize univariate and multivariate outliers. [84]. The question is how to identify an outlier in the dataset?

In the predictive modeling it is necessary to have a dataset in a good shape, without misleading values, nice and clean, that data describe task as adequate as possible. Dealing with outliers can increase performance of model significantly. There are few popular methods for identification of outliers in data: Interquartile Range Method, Standard Deviation Method or Z-Score or Extreme Value Analysis, Linear Regression Models like PCA or LMS or High Dimensional Outlier Detection Methods and others.

Interquartile Range Method is detecting outliers of non-Gaussian distribution by using box in box and whisker plot. The identification of outliers is performed by dividing dataset into four section or quartiles, where box plot itself lies between first and third, which is 25th and 75th percentiles of data, divided by fiftieth percentile as a median. IQR detects values below 1^{st} and above 3^{rd} quartile and based on limit, which is usually of 1.5 times of IQR above or below, in severe cases of 3 times [85].

Popular way of detecting outliers of Gaussian distribution is known as standard deviation method. The outlier is identified if it is lying more k-number of standard deviation from mean. It is usual way how to remove outliers from the rest of dataset. If dataset set is not large in size, we can use 2 standard deviations if it is growing in size, we can apply 3 or 4 standard deviation from mean [86]. There are many other methods, dealing with outliers, depends on task definition, data, dimensionality and distribution.

3.9 Model Selection & Evaluation

To develop a solution for our machine learning task requires long process of preparation, pre-processing, evaluation and tuning of many details along the way. Next part of successful predictive modelling is to find the best possible model for out data. There are certain aspects

which are playing an important role when selection model like performance and accuracy of the model, complexity, interpretability, speed of training or scalability [87].

Fitting a model includes steps which might be responsible for great model performance. There might be many challenges which we need to solve while developing our model. Overfitting, under-fitting, low model performance or inappropriate model selection are few to name. How to find a right way how to handle with possible mentioned challenges. In following section, we will discuss most common and effective methods, which will support our predictive modelling.

3.10 Model Training & Testing

Model training and testing is crucial step towards successful solution for our machine learning task. One of the first steps is to split data on training and testing datasets. It is necessary to keep in mind that several rules must be taken into account when developing a model. Rule number one in model training and testing is to never use testing data set for training. Usual ratio of splitting is 75% of data for training and 25% of data for testing [88]. The reason behind is that if we don't split data, outputs of the model might be biased on new inputs, incorrect results and predictions will not be that well performing.

Training dataset is utilized for training of our prediction model. During the training phase, in supervised learning, a used model is learning from available, pre-processed and most importantly labelled data. On the other side, the objective of testing set is to examine and find out how well is our model doing, how is performing, what is accuracy and so on. In contrast with train set, test set comes without labelled data and is crucial for final evaluation of our model [89].

The aim of splitting data is to find out how our model is developed and if it is performing well. Outcomes which we want to avoid are overfitting and underfitting. In machine learning we say that model is underfitting or oversimplifying when it is not able to find trends in available data and its performance is low. In other words, it means that our model is not right fit for our data. A reason behind underfitting might be for example when the size of dataset is too low. We are referring to overfitting or overcomplicating when our model is capturing every single data point without any error. The model is simply not generalizing data well, rather describing relationships between data and labels too much in training set and failing to predict from new data entries. Crucial for the model is to capture signals from data rather than noise, representing unimportant data [90]. The problem of noise is that our model is gaining knowledge, which is not relevant for prediction. If the model performs very well on training set, doesn't mean it will accomplish same results in testing set. If the model doesn't capture signals but uses noisy data in learning process it will end up with overfitting and poor

predictions. That's how we would know, that our model is either overfitting or underfitting and we need to do further adjustments and corrections on the model. The aim of properly developed model is to distinguish between signals and noise.

3.11 Detecting & Preventing Errors

When our model is not performing well we want to know what is happening and what might go wrong. There are several methods which are helping us to determine what is wrong and what is causing troubles. Let's first discuss more deeply about types of errors. As mentioned before we have two types of errors: over-fitting and under-fitting. These types of errors are based on bias-variance trade-off, which has direct connection to complexity of model. Bias is related to under-fitting, when model is not flexible to describe data well. So, when a model is under-fitting it has high bias but low variance. When it comes to training, under-fitting model has bad results already on training set as well as bad results on test set. Variance is on the other side high when model is overfitting and bias is low. It means that model is reacting sensitively to data. When model is over-fitting, the performance on training dataset is suspiciously high and poor on testing data. A good model is somewhere in the middle. It finds balance and therefore we talk about trade-off. There are few rules which everyone should keep in mind when developing prediction model to avoid over-fitting. First of all, we need to divide dataset into training and testing, cross-validating, trying different algorithms, regularization, fitting parameters and tuning hyper parameters, ensemble method application, getting more data or removing unimportant features [91].

One of the most common way how to identify errors and avoid them is cross-validation technique. Cross-validation is a method utilized for identification of how well is model performing and able to generalize out of data. The procedure is used also as a model selection tool. Thanks to cross-validation we are able to measure model performance more precisely and might help us to solve task of over-fitting. A drawback of such method is that it requires more computational resource and time. In contrast with training-test split of data, we are adding one more: training-validation-testing sets [92]. Following figure shows difference between cross-validation split and train-test split:

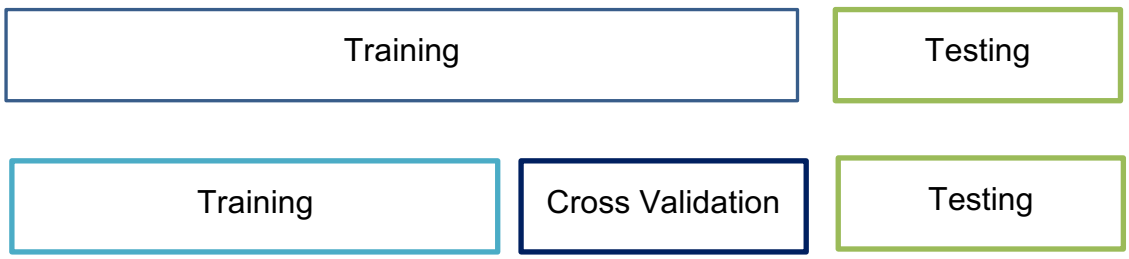


Figure 9 Cross Validation Dataset Split Example

There are more types of cross-validation. The most common type, which is used is called k-fold cross validation. K-fold cross validation works with k number of equally divided data, also known as folds. The K-fold cross validation is an iterative process, where for each iteration another fold is used as validating one, also known as hold-out fold, and the rest is used for training [92]. Another type of cross-validation is Leave-One-Out Cross-Validation or shortly LOOCV. The fundamental part of LOOCV is based on k-fold cross-validation. The difference between these two is that LOOCV uses sum of all observation as a k and each iteration one observation is kept as hold out fold [93].

There are other methods which are helping us to avoid overfitting. As we discussed in previous paragraphs, methods related to feature selection and extraction might be applied to reduce number of features and make model simpler.

Another method, often used is called model tuning. The aim of this method is to increase performance of our model. It might be achieved by tuning of settings or configurations, often referred as parameters. Parameters are settings of variable, which are characteristic for model itself and might be concluded out of available data. Examples of parameters are coefficients or weights in algorithms. Parameters are also depending on type of algorithm and might differ from one to another. The aim of model tuning is to detect best possible parameters [94].

Closely related and often interchanged is tuning of hyper-parameters of estimator. Hyper-parameters are configurations of model which are not internal for prediction model and it is not possible to be concluded directly from data. They have to be provided to the model. There are several approaches, which are trying to solve a task of parameters detection. Common tuning methods are known as Grid Search Parameter Tuning or Random Search Parameter Tuning [95].

3.12 Evaluation Metrics Selection

One of the final parts of prediction model development is evaluation. To know how our model is performing is crucial for decision making regarding next process steps. As everywhere else, also when developing a model, we want to know how it is working before we release it into production. We want to know how well it is. There are several approaches and metrics, which are helping us gain knowledge about performance of our model. Metrics are used in different manners, depending on the model itself. Not every metric might be used or is effective for every type of algorithm. There are different evaluation approaches to supervised tasks of regression and classification as well as unsupervised clustering tasks. For our case, only metrics of supervised learning are relevant, and we will pay attention on them.

In supervised learning, regression tasks, which are focused on continuous value predictions, there following common methods to examine performance of model: Root Mean Squared Error, R Squared, Mean Absolute Error and Mean Squared Error. Starting with Room Mean Squared Error or shortly RMSE as one of the most utilized evaluation techniques calculates a square root of the mean residual's distance from regression line. With statistical words we can say that root mean squared errors is a metrics, calculating standard deviation of estimated errors from regression line. In simple explanation we can say that RMSE describes how far are estimations from best model [96].

Another commonly applied method for regression model evaluation is called R squared or shortly R^2. We can refer to it as coefficient of determination as well. It is a metric, telling how distant the points are from the fitted regression [97]. R^2 estimates score in range of 0 and 1, where the results closer to 0 are not very well and results close to 1 represents good fit. But this evaluation metric has also limitation. It means that low results don't have to necessarily describe a bad model and vice versa. If the score is low but there are significant variables, meaning the changes in them would represent a change in target variable, we can still make an outcome out of it [98]. Mean Absolute Error or shortly MAR as from the name is clear is the mean of the absolute value of the errors and Mean Squared Error or MSE is the mean of the squared errors [99]. In classification tasks we have following evaluation metrics: confusion matrix, accuracy, logarithmic loss function, and others.

4 Results

4.1 Data Exploration & Pre-processing

In the following chapter we will focus on understanding of our dataset, its shape, content and relationships between individual features. EDA examines different characteristics of available dataset, looks on the data from different point of views and aims to find correlations between most significant variables in the data. Pre-processing helps to get the data ready for predictive modelling. The individual tasks are done with Python. Following objectives are met:

- Dataset exploration and description – understand the data and gain maximum insights
- Identifying missing values – find variables with extensive amount of missing values
- Visualization of geospatial data on maps – to get the grasp about spatial variables
- Univariate analysis – understand target variable
- Multivariate analysis – understand underlying relationships between each feature of data set and dependent variable
- Dataset preparation and pre-processing – tidy and prepare data for comprehensive analysis and predictive data modelling

The objective of this chapter is to provide better understanding of the data, its variables, and the correlation of independent variables to dependent variable. Outcomes will server for further project steps.

4.2 Data Description

The dataset which is utilized for our research comes from independent, non-commercial project called Inside Airbnb[1]. The objective of the project is to provide data to community of specialists for better understanding of Airbnb listings [12]. We will use the data for prediction model to create a source for better price estimation from hosts as well as guests side.

We will work with Los Angeles dataset, uploaded on 02 May 2017. Comprehensive data analysis and exploration is conducted in Jupyter Notebook, with Python, more specifically

[1] http://insideairbnb.com/get-the-data.html

with following packages: Pandas, Matplotlib, Numpy and Seaborn. First step is to import needed libraries and load the file:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

```
#read csv file
LA_DATA = pd.read_csv('/Users/Filip/Downloads/LA.csv')
```

```
#show a shape of dataset
LA_DATA.shape
```

```
(31253, 95)
```

Figure 10 Data Loading

After shaping, we see that the file, which we will use – listings.csv, contains 95 variables (94 features and 1 target variable-price) with 31,253 records. Not all of them are relevant, usable or reliable for our research and model development, therefore we will need to conduct feature selection. Following analysis shows available variables:

```
#check available features
LA_DATA.columns
```

```
Index(['id', 'listing_url', 'scrape_id', 'last_scraped', 'name', 'summary',
       'space', 'description', 'experiences_offered', 'neighborhood_overview',
       'notes', 'transit', 'access', 'interaction', 'house_rules',
       'thumbnail_url', 'medium_url', 'picture_url', 'xl_picture_url',
       'host_id', 'host_url', 'host_name', 'host_since', 'host_location',
       'host_about', 'host_response_time', 'host_response_rate',
       'host_acceptance_rate', 'host_is_superhost', 'host_thumbnail_url',
       'host_picture_url', 'host_neighbourhood', 'host_listings_count',
       'host_total_listings_count', 'host_verifications',
       'host_has_profile_pic', 'host_identity_verified', 'street',
       'neighbourhood', 'neighbourhood_cleansed',
       'neighbourhood_group_cleansed', 'city', 'state', 'zipcode', 'market',
       'smart_location', 'country_code', 'country', 'latitude', 'longitude',
       'is_location_exact', 'property_type', 'room_type', 'accommodates',
       'bathrooms', 'bedrooms', 'beds', 'bed_type', 'amenities', 'square_feet',
       'price', 'weekly_price', 'monthly_price', 'security_deposit',
       'cleaning_fee', 'guests_included', 'extra_people', 'minimum_nights',
       'maximum_nights', 'calendar_updated', 'has_availability',
       'availability_30', 'availability_60', 'availability_90',
       'availability_365', 'calendar_last_scraped', 'number_of_reviews',
       'first_review', 'last_review', 'review_scores_rating',
       'review_scores_accuracy', 'review_scores_cleanliness',
       'review_scores_checkin', 'review_scores_communication',
       'review_scores_location', 'review_scores_value', 'requires_license',
       'license', 'jurisdiction_names', 'instant_bookable',
       'cancellation_policy', 'require_guest_profile_picture',
       'require_guest_phone_verification', 'calculated_host_listings_count',
       'reviews_per_month'],
      dtype='object')
```

Figure 11 Available Columns

Based on the first observation we see that many variables will hardly correlate with price. URL related features, naming conventions or information about verifications will not contribute to prediction model. Based on the rational decision and intuitive thinking we will drop most unrelated variables to make further data processing smoother. Observation also uncovered that there are many features related to geographical location of listings – country, city, state, zip code, street or longitude and latitude. The aim is to simplify geographical feature role and understand which of these features are the best suited for our model. There is no need for country, state or city columns as we are working with Los Angeles data and will suit the model for this area. Another set of variables is identification information about host or listing itself, which might not be needed directly for our prediction model but might be used as a reference.

Prediction model will also not work properly with variables, which have higher rate of missing values and therefore these will be dropped. Following analysis shows percentage of missing values per variable:

```
#check and sum all missing data from all variables
pd.isnull(LA_DATA).sum(axis=0)

#reflect missing values in percents
series = pd.isnull(LA_DATA).sum(axis=0)* 100 / len(LA_DATA)

#plot a histogram in descending order
series.sort_values(ascending=False).plot(kind='bar', fontsize=14, figsize=(22, 8), color= 'red')
plt.xlabel('Variables', fontsize = 18)
plt.ylabel('Missing values in %', fontsize = 18)
```

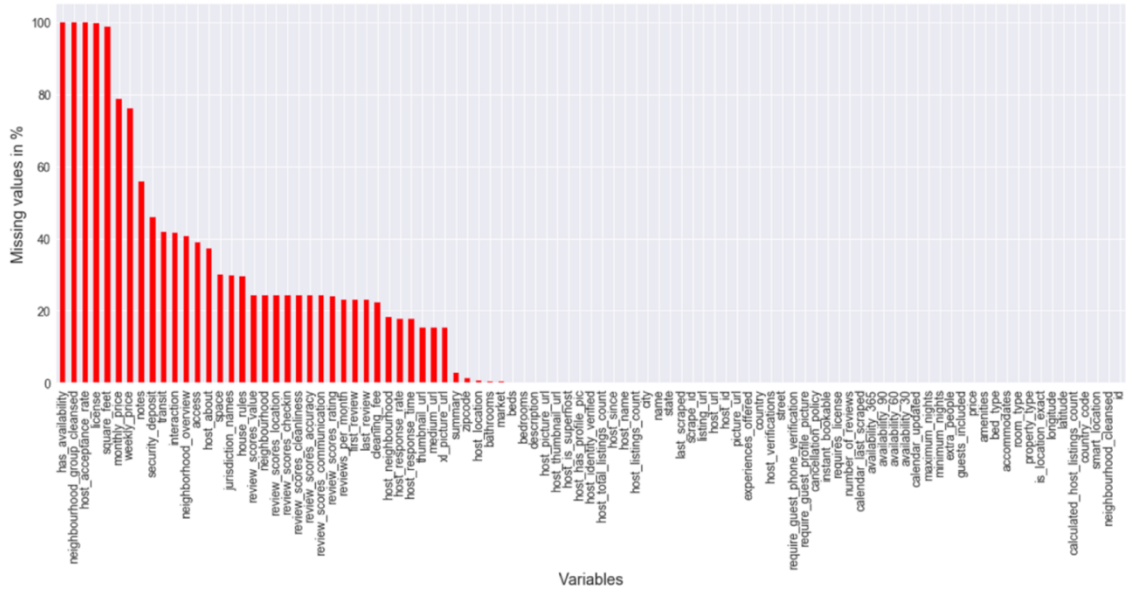


Figure 12 Missing values - Initial Analysis

The analysis showed, that there are many variables with high percentage of missing values like: neighbourhood, weekly price, monthly price and others. The variables with high rate of

missing values will be dropped as the prediction model would not provide expected results. Moreover, there are many features, which simply doesn't contain enough information from prediction point of view. Based on intuitive thinking and missing value analysis we have selected following variables for further analysis of correlations with the target variable 'price':

```
columns = ['price','host_since', 'host_response_time', 'host_response_rate',
           'host_is_superhost', 'host_identity_verified', 'host_total_listings_count',
           'neighbourhood_cleansed', 'zipcode', 'smart_location', 'latitude', 'longitude',
           'is_location_exact', 'property_type', 'room_type', 'accommodates', 'bathrooms',
           'bedrooms', 'beds', 'bed_type', 'amenities', 'cleaning_fee', 'guests_included',
           'extra_people','maximum_nights', 'minimum_nights', 'availability_30',
           'availability_60', 'availability_90', 'availability_365', 'number_of_reviews',
           'review_scores_rating','review_scores_accuracy', 'review_scores_cleanliness',
           'review_scores_checkin', 'review_scores_communication', 'review_scores_location',
           'review_scores_value', 'reviews_per_month', 'description', 'calendar_updated',
           'calendar_last_scraped','instant_bookable']
```

```
LA_DATA_v2 = pd.read_csv('/Users/Filip/Downloads/LA.csv', usecols=columns)
LA_DATA_v2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31253 entries, 0 to 31252
Data columns (total 43 columns):
description                    31238 non-null object
host_since                     31239 non-null object
host_response_time             25692 non-null object
host_response_rate             25692 non-null object
host_is_superhost              31239 non-null object
host_total_listings_count      31239 non-null float64
host_identity_verified         31239 non-null object
neighbourhood_cleansed         31253 non-null object
zipcode                        30858 non-null object
smart_location                 31253 non-null object
latitude                       31253 non-null float64
longitude                      31253 non-null float64
is_location_exact              31253 non-null object
property_type                  31253 non-null object
room_type                      31253 non-null object
accommodates                   31253 non-null int64
bathrooms                      31156 non-null float64
bedrooms                       31218 non-null float64
beds                           31194 non-null float64
bed_type                       31253 non-null object
amenities                      31253 non-null object
price                          31253 non-null object
cleaning_fee                   24279 non-null object
guests_included                31253 non-null int64
extra_people                   31253 non-null object
minimum_nights                 31253 non-null int64
maximum_nights                 31253 non-null int64
calendar_updated               31253 non-null object
cleaning_fee                   24279 non-null object
guests_included                31253 non-null int64
extra_people                   31253 non-null object
minimum_nights                 31253 non-null int64
maximum_nights                 31253 non-null int64
calendar_updated               31253 non-null object
availability_30                31253 non-null int64
availability_60                31253 non-null int64
availability_90                31253 non-null int64
availability_365               31253 non-null int64
calendar_last_scraped          31253 non-null object
number_of_reviews              31253 non-null int64
review_scores_rating           23726 non-null float64
review_scores_accuracy         23689 non-null float64
review_scores_cleanliness      23688 non-null float64
review_scores_checkin          23661 non-null float64
review_scores_communication    23690 non-null float64
review_scores_location         23660 non-null float64
review_scores_value            23648 non-null float64
instant_bookable               31253 non-null object
reviews_per_month              24030 non-null float64
dtypes: float64(14), int64(9), object(20)
memory usage: 10.3+ MB
```

Figure 13 Initial Feature Selection

Based on the selected variable exploration, there are 42 features and 1 target variable – price. We can sum up that they are of data type: float, integer and object. After a closer look into data, we have seen that numerical features are simple values, representing number of rooms, bathrooms, how many listings is owner offering and others. There are also Boolean features, where true and false are values: 'is_location_exact', 'instant_bookable' or 'extra_people'. Categorical variables are the ones without a linear dependency like: 'property_type', 'room_type', amenities or 'neighbourhood_cleansed'. Last category is especially for description variable, which is text analysis feature, containing sentences or longer text. The dataset still contains lot of missing data. At this stage of research, we will consider all variables above and will analyse them more deeply to find most correlating ones.

4.3 Visualization of Geospatial Data

Visualization of data is crucial for better understanding and gaining a big picture perspective over available features and records. For geospatial description of our dataset, we will Tableau. Tableau is an industry leading tool which is applied for data visualization, data discovery and analytics [100]. For the geolocation we will refer to Zip Code. Following map shows us number of available rentals per Zip Code location:

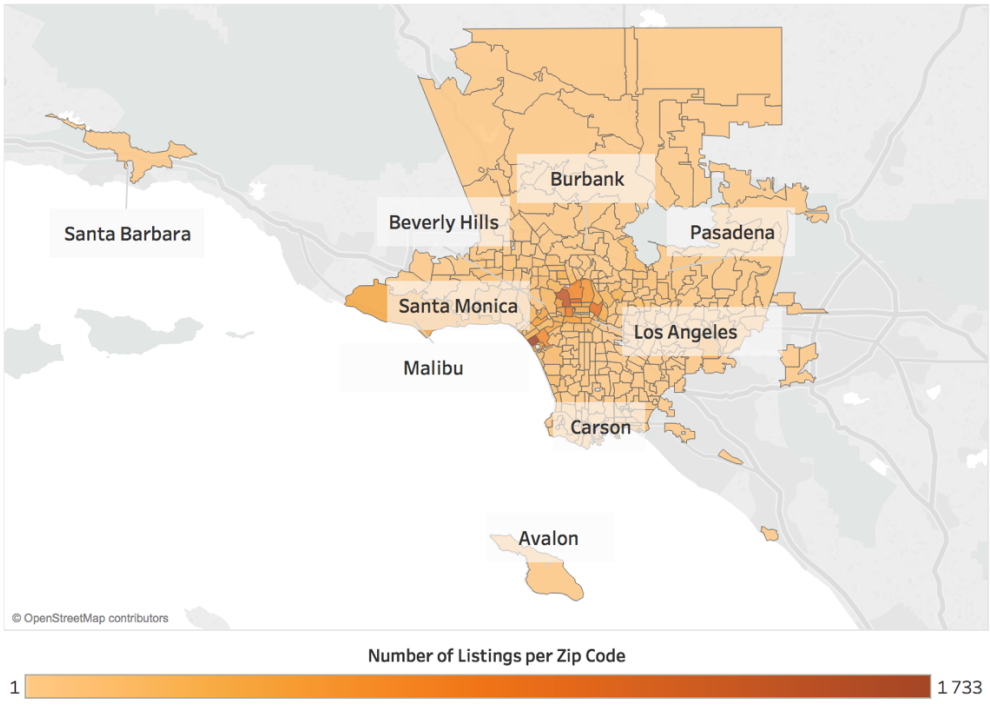


Figure 14 Geospatial Visualization - Listing Distribution per Zip Code

Based on the first assumption we can observe that specific, more central, coastal or recreational locations have higher number of listings, than the ones in suburbs. We assume

that more recognized places will have higher demand for accommodation, therefore the price will be higher, renting will be more profitable, and also more host will be willing to rent. The places like Malibu, Santa Monica, Down Town or Hollywood are definitely more searched and lucrative places in Greater Los Angeles Area. The question is how this fact drives price of rentals? Following geospatial visualization shows distribution of average price for each zip code:

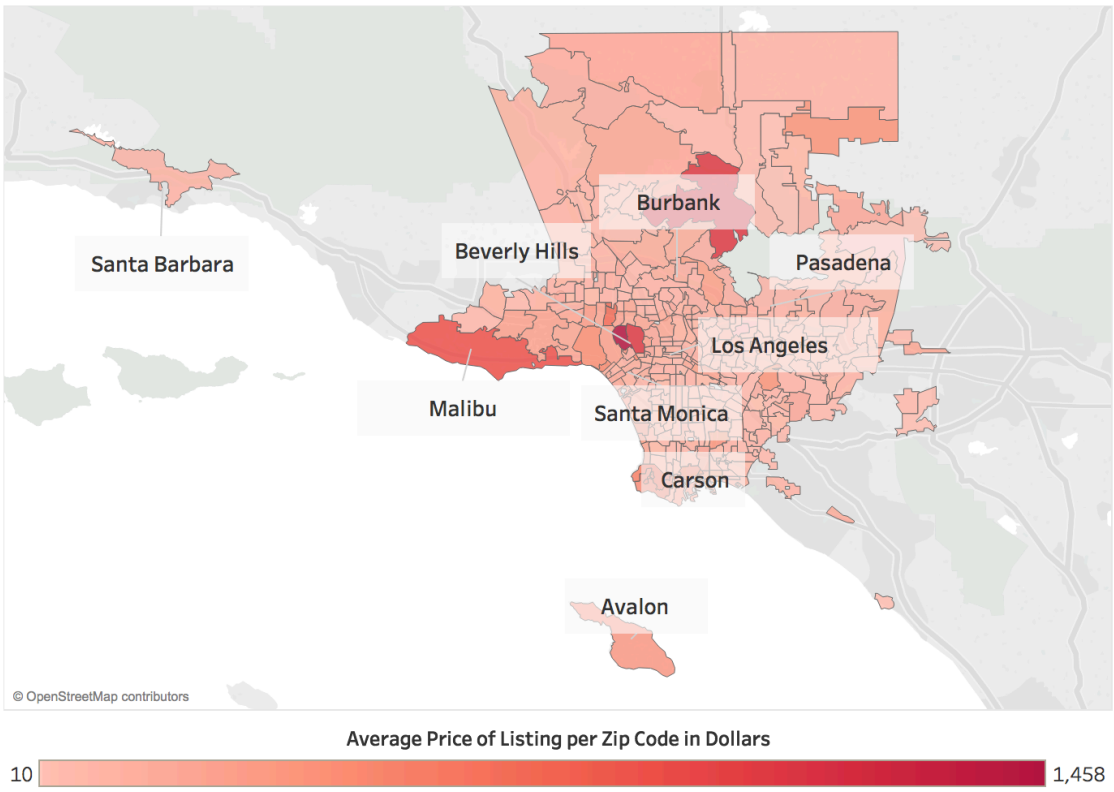


Figure 15 Geospatial Visualization - Average Price per Zip Code

The figure uncovered few interesting facts. Based on our previous assumption that the demand should drives price higher was partially right for some locations like Malibu or Beverly Hills. On the other side the average prices in downtown is not higher than in the rest of the city. Based on the gained knowledge, there is not direct correlation between price of the listing and zip code.

There must be also other features, which correlates with price more than location and zip code itself. We will also consider in further work to merge Zip Codes into neighbourhoods and analyse if it will increase performance of our models. Currently based on observation we should also consider as feature influencing price popularity and lucratively of zip code based on some specific significance of that area – university availability, sightseeing, seaside or other price driving factors. We assume that there are other factors, more correlated with price. We will see in further analysis.

4.4 Target Variable Analysis

Standard way how to understand behavior of any variable is to conduct a univariate analysis. Univariate analysis is the most fundamental form of creating a big picture about data [101]. It is not focusing on the underlying relationships between variables but describes and dive deep into single feature. In our case, we will focus on 'Price' column as target variable. Following analysis shows feature description:

```
LA_DATA['price'].describe()
```

```
count       31253
unique        703
top       $100.00
freq         1028
Name: price, dtype: object
```

Figure 16 Initial Data Description

Out of the description above there is a clear understanding that every single entry contains price indication and there is no missing value. Price has 703 unique values but as description showed, type of our variable is 'object'. It is not good data type for price. For statistical description we would prefer to have price as numerical value. Following step transforms variable from 'object' type to 'float' type and description is conducted again:

```
LA_DATA['price'] = (LA_DATA['price'].str.replace(r'[^-+\d.]', '').astype(float))
```

```
LA_DATA['price'].describe()
```

```
count    31253.000000
mean       180.094039
std        418.502140
min         10.000000
25%         69.000000
50%        100.000000
75%        170.000000
max      10000.000000
Name: price, dtype: float64
```

Figure 17 Price Transformation and Description

After transformation, data type 'price' is changed to 'float' with range between minimum listing price $10.00 and maximum $10 000.00. The average price of listing per night in Los Angeles is $180.00. Taking into account the minimum price, maximum price and the average we can expect strong skewness. As the most listings are in the range from 10.00 to 170.00 we have to consider a lot of outliers. Following figure shows distribution of price with outliers:

```
sns.set(font_scale=2)
sns.mpl.rc("figure", figsize=(20,10))
sns.distplot(LA_DATA_v2['price'])
```

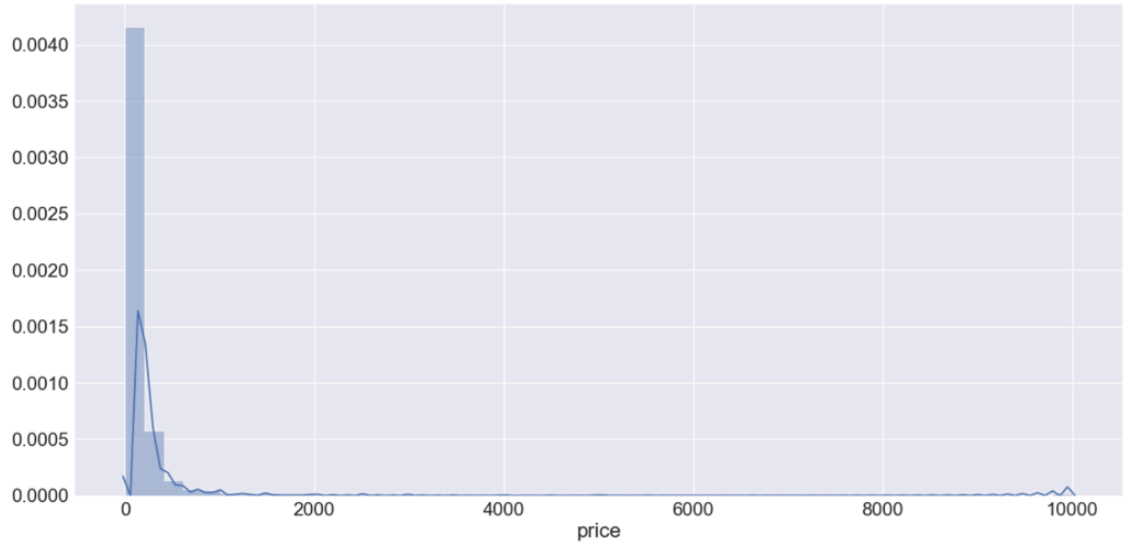


Figure 18 Price Distribution

Distribution visualization shows that target variable “price” deviates from normal distribution and is highly skewed to the right side – the skewness is highly positive. Analysis of skewness and Kurtosis:

```
print("Skewness: %f" % LA_DATA_v2['price'].skew())
print("Kurtosis: %f" % LA_DATA_v2['price'].kurt())
```

```
Skewness: 14.262443
Kurtosis: 277.499306
```

Figure 19 Price Skewness & Kurtosis

As we said it is positively skewed and peaky, possibly with outliers. We will utilize feature transformation further on. So far, we will analyse variable as is. There is rational explanation why the price is so skewed. First of all, the Airbnb is aimed specially for short term renting and we need to distinguish between types of offered accommodation. In our dataset there are listings with shared rooms, private rooms, apartments or even whole houses. Price for a whole house is much higher than price for a shared room. Therefore, transformation or outlier removal is required. We will describe this step in feature transformation part.

4.4.1 Numerical Features Analysis

In the following section we will have a closer look on individual numeric features, their characteristics, their distribution and their relationship to our target variable: price. The aim of this section is to detect the ones with the strong correlation in regard to price of Airbnb listings. Later on, we will have a look on categorical features as well. To get a deep understanding of what we have in the data, we are beginning with the visualization of individual feature distribution. Following figure shows distribution of our dependent variable:

```
#Following function help us to get numerical features.
def feature_types():

    num_f = LA_DATA_v2.select_dtypes(include=['int64','float64']).columns
    num_f = num_f.drop(['price'])
    return list(num_f)

num_f = feature_types()
```

```
#Plotting distribution of numerical variables
sns.set(font_scale=4)
features = pd.melt(LA_DATA_v2, value_vars=sorted(num_f))
grid = sns.FacetGrid(features, col= 'variable', size = 12, col_wrap=3,
                     sharex=False, sharey=False)
grid = grid.map(sns.distplot, 'value')
```

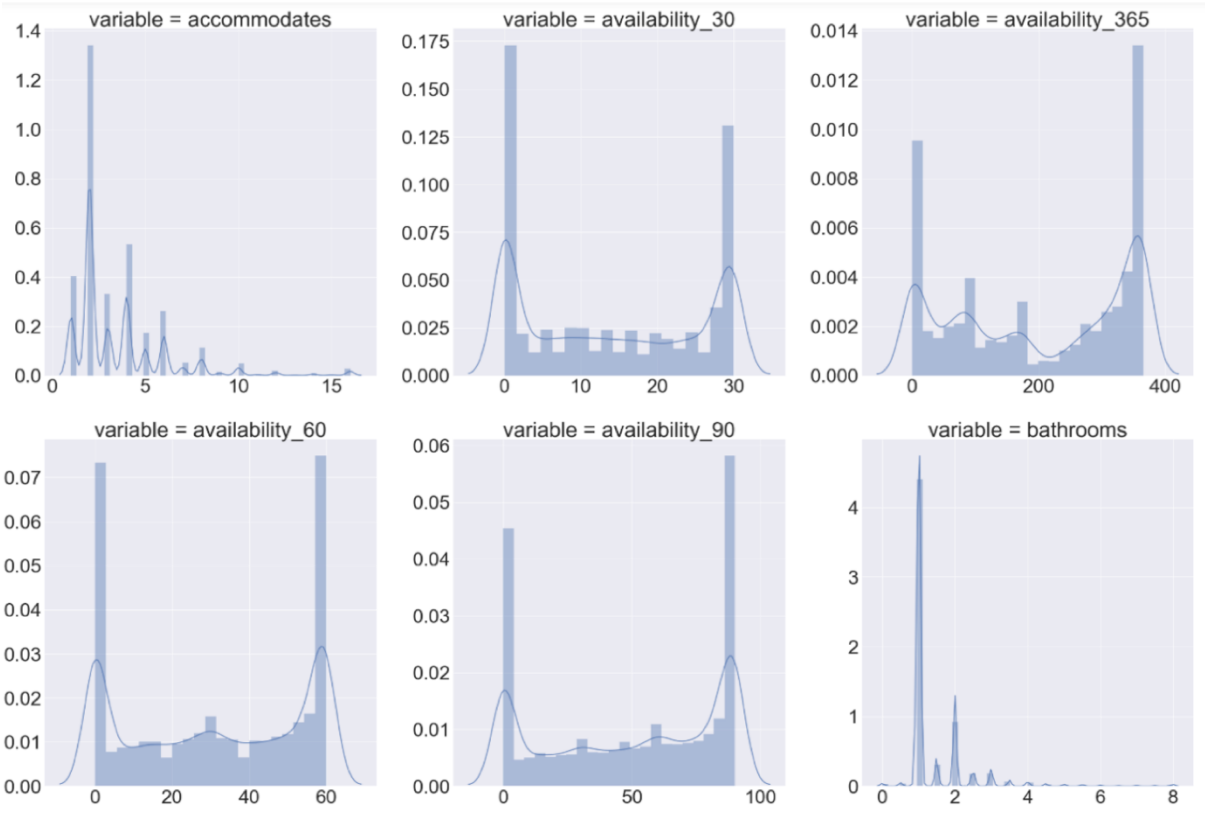


Figure 20 Univariate Analysis 1

As we can see from the results above, there some variables, which are availability related to each other. All of these variables are showing same distribution with slight differences. The most occurred values are close to zero, so fully booked, or on the other side close to be fully available. All the variables with 0 values are not very pleasant and might represent challenge. We need to consider next steps regarding such variables.

When we have a look on accommodates or bathroom count, we see again positive skewness as we had before with our target variable. We need to consider further transformation, but as some of such values might represent count values, it is necessary to analyse it further. Continuing with distribution analysis of the rest numerical features, we have following results:

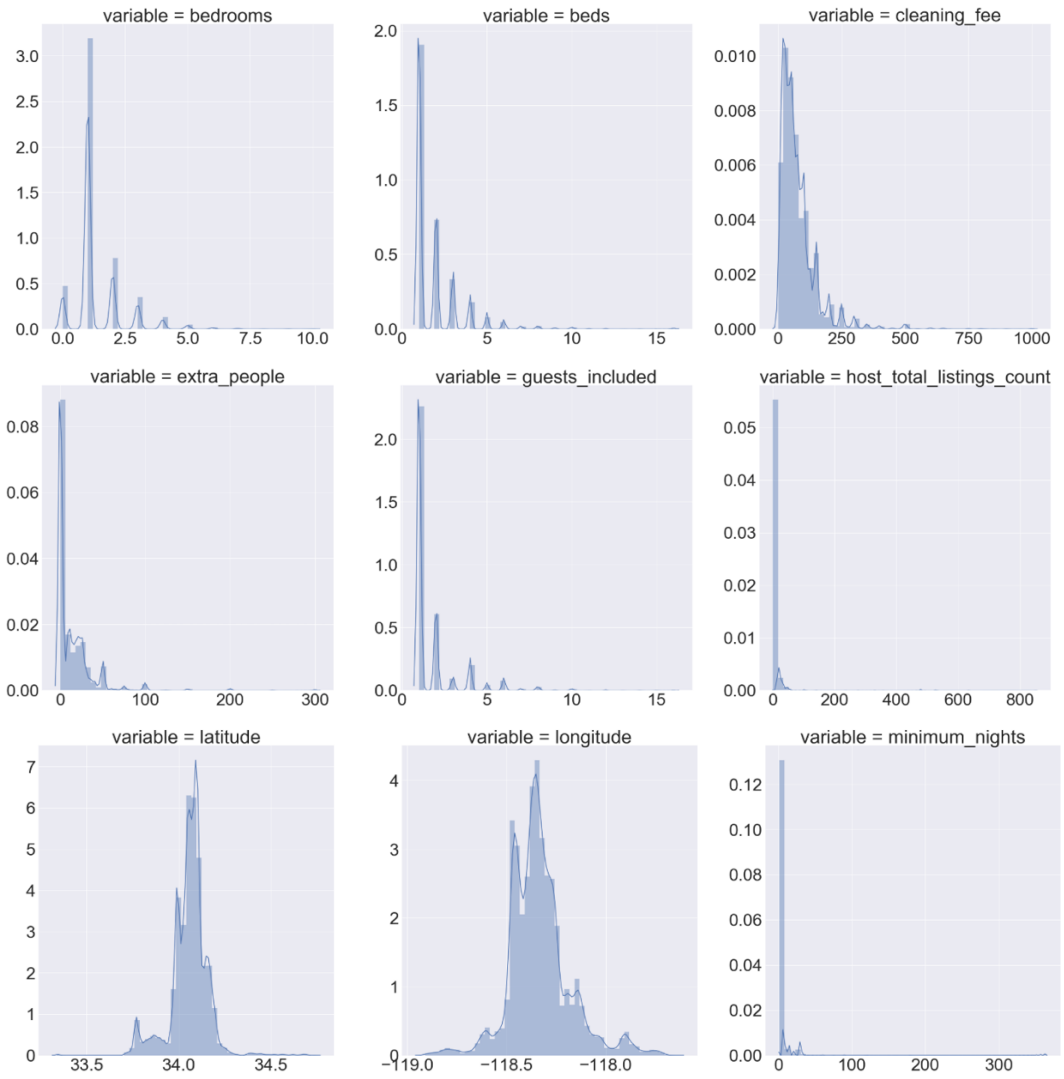


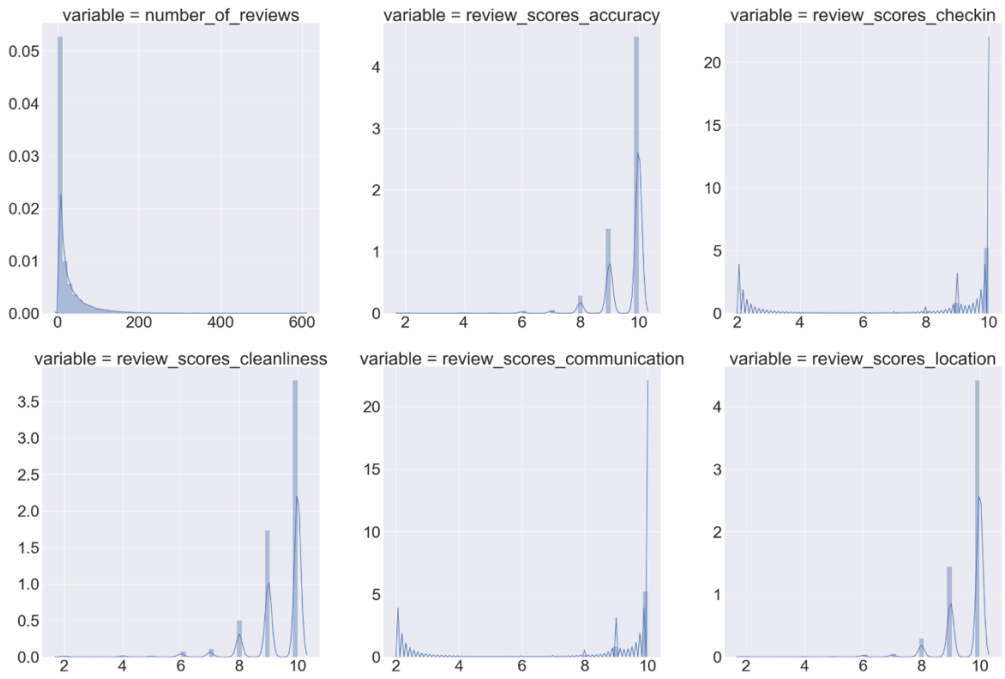
Figure 21 Univariate Analysis 2

Again, we can see similar outcomes as we had before, many variables are positively skewed and possible feature transformation is in the game. Variables of ‘bedrooms’, ‘beds’, ‘guests_included’ are very similar in distribution. We can see there two variables, related to

price - ‘cleaning_fee’, ‘extra people’. They are strongly positively skewed. Distribution is very similar to our target variable.

Another strongly positively skewed feature, ‘host_total_listings_count’, contains extreme values. It might be understood that most of the host have 1 of few listings, we assume. Airbnb has also recently implemented an API, which should support hosts, to administrate their listings simpler in case they have more of them. There are few hosts which are having much more listing, that regular people. With the further research we have found out that these extreme cases of Airbnb users are called mega hosts, renting more than 1000 properties at the same time. They might represent outliers as some of them are managing hotels, hostels or real estate companies. The trend of mega hosts is increasing but smaller hosts are still in majority [102]. We need to tackle this feature in order to gain more relevant insights.

Another couple of variables represents the geographical location of our variables. We can see there more less normal distribution with some areas more frequent then others and some listing standing isolated. As we could see before, form geospatial analyses, city center is having most of the listings, as well as frequent recreational places like Santa Monica and other coastal areas. Following analysis shows the rest of numerical features, mostly related to review score:



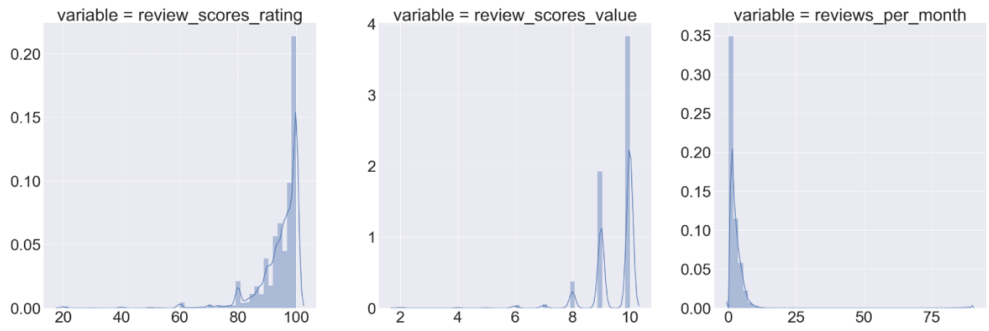


Figure 22 Univariate Analysis 3

We can see very similar pattern in the review related variables: 'review_scores_accuracy', 'review_score_checkin', 'review_scores_communication', 'review_scores_value', 'review_scores_location', 'review_scores_cleanliness' and 'review_scores_rating'. Same high negative skewness.

Interesting results are shown in analysis of ‘minimum nights’, where we can see huge number of zero values, but also wide range of extreme values, where minimum stay is more than 300 nights. We will have a closer look on this variable and try to find reasons behind such outlying values.

In another two variables, ‘number_of_reviews’ and ‘reviews_per_month’ we see strong positive skewness. In most numerical values we can observe positive skewness where further data transformation is required to have possible better prediction model results.

Let’s have a look on the variables, which showed us some extraordinary results of analysis, where we need to know more about data background. We will examine: ‘minimum nights’, ‘host_total_listings_count’ ‘number_of_reviews’:

```
LA_DATA_v2['minimum_nights'].describe()
```

```
count    31253.000000
mean         3.208460
std          8.169057
min          1.000000
25%          1.000000
50%          2.000000
75%          3.000000
max        365.000000
Name: minimum_nights, dtype: float64
```

Figure 23 Minimum Nights Description

As we expected, there are with high probability outliers. Most of values are up to 3 minimum nights.

```
LA_DATA_v2['number_of_reviews'].describe()
```

```
count    31253.000000
mean        20.842543
std         38.106641
min          0.000000
25%          1.000000
50%          6.000000
75%         23.000000
max        605.000000
Name: number_of_reviews, dtype: float64
```

Figure 24 Number of Reviews Description

The same applies to 'number_of_reviews', where most of the values has 23 reviews and average number of reviews is 21.

```
LA_DATA_v2['host_total_listings_count'].describe()
```

```
count    31239.000000
mean         5.171132
std         24.143834
min          0.000000
25%          1.000000
50%          2.000000
75%          4.000000
max        855.000000
Name: host_total_listings_count, dtype: float64
```

Figure 25 Host Total Listings Count Description

When we have a look on total listings, there are in average 5 listings per one host, which might be misleading due to max values of 855 listings per one host. All these outliers, possibly as a results of novelty data or normal outliers might be causing challenges and we need to process these variables further to work with them in next steps of our model development. In the next section we will have a look on categorical variables.

4.4.2 Categorical Features Analysis

Our dataset contains features, which are not represented by integers but rather by strings or time-series and they are in object data type. We refer to them as categorical variables and believe that they might have a relationship to our target variable – price. Starting with Boolean variables: 'host_is_superhost', 'host_identity_verified', 'is_location_exact', 'instant_bookable':

```
#Following function help us to get categorical features.
def feature_types_cat():

    cat_f = LA_DATA_v3.select_dtypes(include=['object']).columns
    return list(cat_f)

cat_f = feature_types_cat()
```

```
sns.set(font_scale=1.3)
features = pd.melt(LA_DATA_v3, value_vars=sorted(cat_f))
grid = sns.FacetGrid(features, col='variable', col_wrap=4,
                     sharex=False, size = 4, sharey=False)
plt.xticks(rotation='vertical')
grid = grid.map(sns.countplot, 'value')
[plt.setp(ax.get_xticklabels(), rotation=90) for ax in grid.axes.flat]
grid.fig.tight_layout()
plt.show()
```

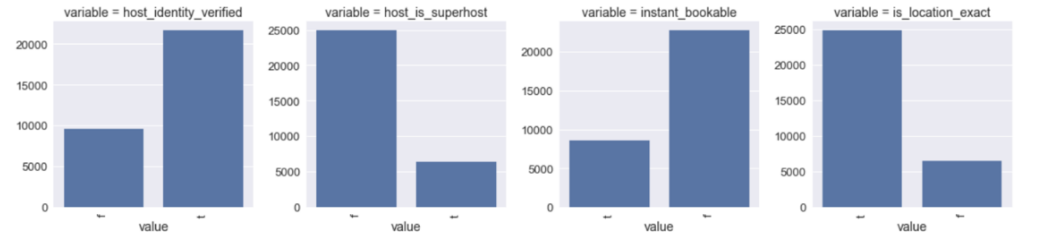


Figure 26 Categorical Feature Analysis 1

Based on the analysis, we have received 4 different distributions of Boolean variables. They are consisting of only two values: True of False. We can see that the total number of values in all of them is more less the same, with range up to 25000. When talking about categorical variables, we need to consider a change to ordinal values if possible. Some algorithms are not able to tackle with categorical values as humans and are not able to extract meaningful information. As an example, is SVM algorithm, which works only with numerical variables. Next in the pipeline are following categorical variables: 'host_response_time', 'room_type', 'bed_type':

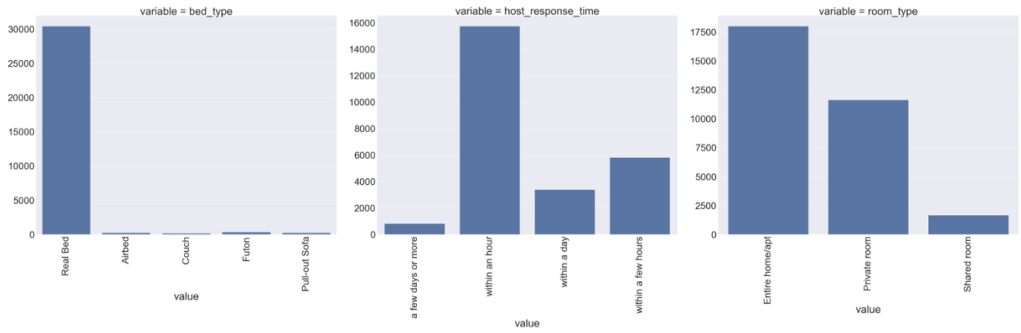


Figure 27 Univariete Analysis 2

The categorical variables are alluring in context to price prediction as they might hide interesting information. It is necessary to note that some of the features, like ‘bed_type’, where variable is over described by one value, in this case ‘Real Bed’. On the other side, with this in mind we can identify extreme values or outliers. As we can see, host_response_time is consisting out of 4 simple values – ‘within an hour’, ‘within a few

hours', 'within a day' and 'within a few days or more'. The same applies to 'room_type' variable, where values are: 'entire hope/apt', 'private room' and 'shared room'. We have mentioned before that for such variables it is good idea to consider transformation into discrete quantitative variables Except longitude and latitude, we have here other location related features, which might play also a significant role in our model. Possibly we will use only one type of geolocation coding to avoid complexity of model and simplify interpretability. Following analysis shows neighbourhood cleansed variable and its descritpiton:

```
df_loc = LA_DATA_v2[['zipcode','smart_location', 'neighbourhood_cleansed']]
df_loc.describe()
```

	zipcode	smart_location	neighbourhood_cleansed
count	30858	31253	31253
unique	310	376	254
top	90291	Los Angeles, CA	Venice
freq	1733	19395	2085

Figure 28 Geospatial Variables Description 1

```
df_loc.head()
```

	zipcode	smart_location	neighbourhood_cleansed
0	93510	Acton, CA	Acton
1	93550	Palmdale, CA	Acton
2	93510	Acton, CA	Acton
3	93510	Acton, CA	Acton
4	90007	Los Angeles, CA	Adams-Normandie

Figure 29 Geospatial Variables Description 2

Looking at the location variables, we understand that there are many unique values in all three of them. Based on the detailed look we have seem many duplicates in the, especially in 'smart_location' and 'zip code'. There are many typos as well as spelling mistakes in them or commas. Cleansed neighbourhood seems to be ok. We will need to correct it. Also, worth to mention is that there are even values with Asian or Russian letters, possibly representing outliers. We also noticed that 'smart_location' as well as 'neighbourhood cleansed' seem to be almost one-to-one. We will also consider possible data transformation of zip code to neighbourhood or vice versa to reduce number of features. Next group of variables are representing time-series related features: 'host_since' and 'calendar_updated':

```
df_time = LA_DATA_v2[['host_since', 'calendar_updated']]
df_time.describe()
```

	host_since	calendar_updated
count	31239	31253
unique	2574	64
top	2014-09-29	today
freq	66	6133

```
df_time.head()
```

	host_since	calendar_updated
0	2016-01-12	5 months ago
1	2015-09-22	2 months ago
2	2016-07-19	2 weeks ago
3	2016-05-12	5 days ago
4	2015-08-16	15 months ago

Figure 30 Time Series Variables Description

Interesting to observe both values, as they might play a role when estimating price. First assumption is that hosts who are longer offering their services with Airbnb might have better estimation of price. Following analysis shows distribution of ‘calendar_updated’ variable:

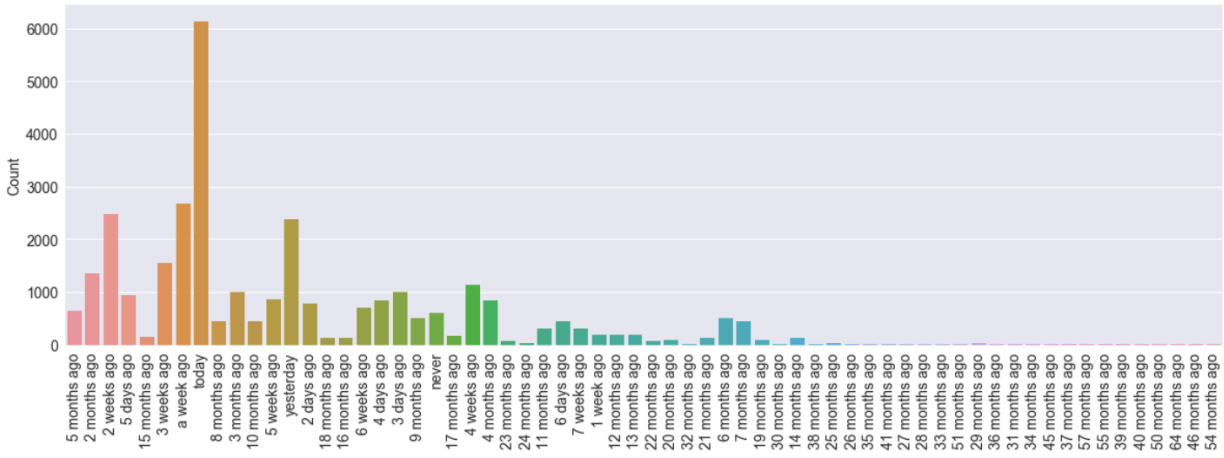


Figure 31 Calendar Variable Values Distribution

For both variables there are many unique values. There might be applied one of the methods of feature engineering – task of feature representation, where we can change format or representation of mentioned variables. It might me much more useful to group hosts based on month and year, or updated calendar within last month, last week, etc. As we can see, many values are within an hour, with most values of today. We will further consider how to transform variables. Airbnb is popular by its wide range of housing types, where you can stay. Last but not least a variable of ‘property type’:

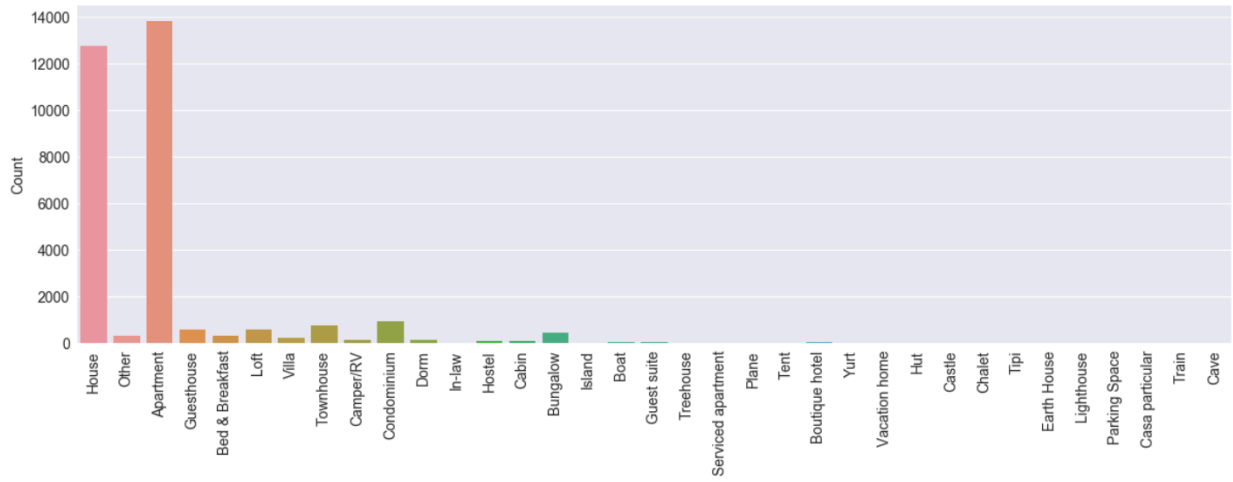


Figure 32 Types of Accommodation Distribution

There is a clear understanding that not all of the types of accommodations are usual and might be misleading like: cave, train, plane, tent or parking space. Mentioned values will help us to detect outliers as they will potentially destroy our model. Obviously most frequent types are apartments or houses.

4.4.3 Initial Feature Transformation

Before we jump in on multivariate analysis and correlation between individual features and their relationships, we need to conduct transformation of data. As we mentioned before, there are few variables, which might improve performance of the models and show us stronger relationship to our target variable after transformation. In the following section we transform required features.

As we mentioned before, price variable needs transformation due to strong positive skewness. A symmetric distribution is always better and, in our case, where skewness is highly positive, we need to conduct a transformation. Normal distribution is optimal for most of the prediction methods. As the price doesn't contain any 0 values, any negative values and skewness is positive, we will apply log transformation, to compare how it behaves in relationship to other variables. as follows:

```
LA_DATA_v2['log_price'] = LA_DATA_v2['price'].apply(lambda x: math.log(x))
```

Figure 33 Price Log Transformation

By log transformation we are aiming to reduce skewness. Following figure shows, how has the distribution of price has changed after log transformation:

```
sns.mpl.rc("figure", figsize=(20,10))
sns.distplot(LA_DATA_v2['log_price'])
```

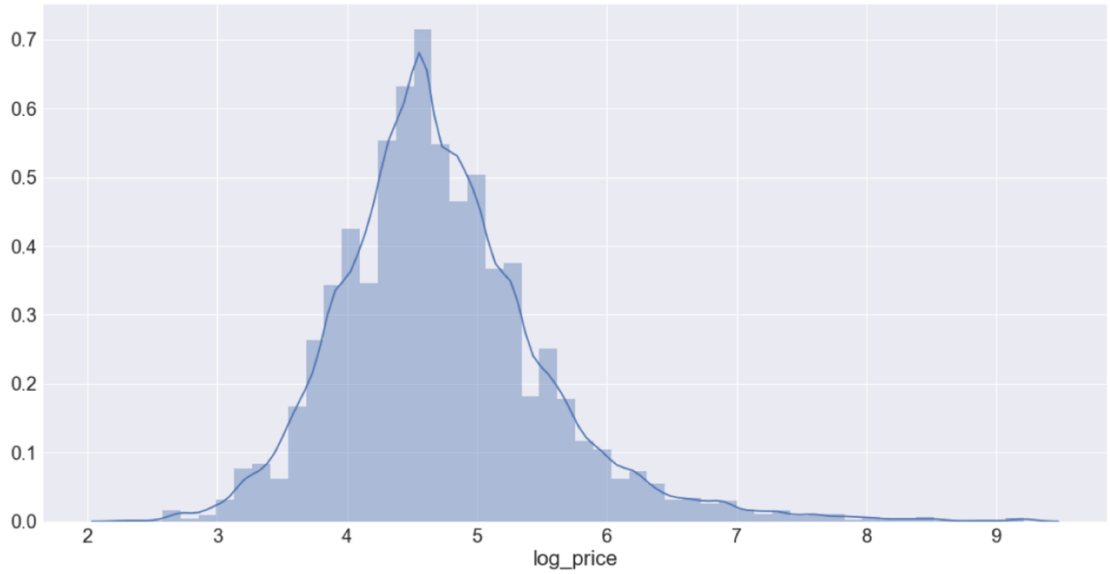


Figure 34 Log Price Transformation Distribution

As we can see the result is much better, but we still have here outliers. Let's compare log price with price:

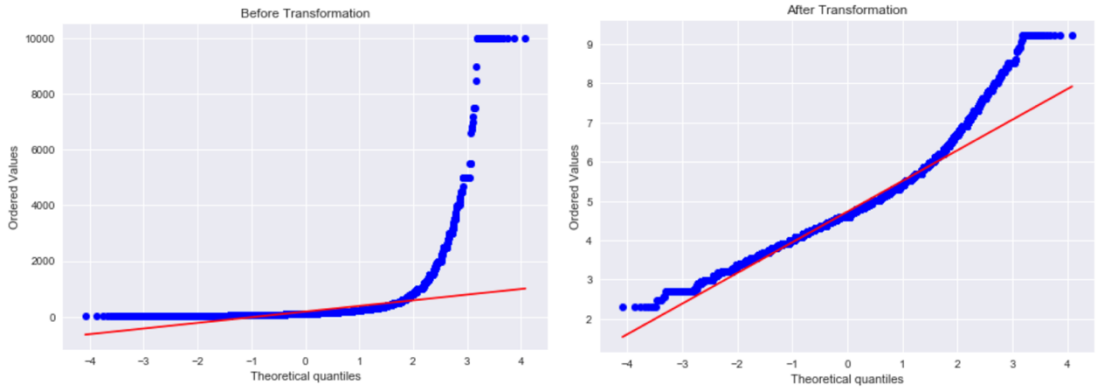


Figure 35 Price Quantiles

We can observe that the price after log transformation is much closer to the normal distribution than before. As we can see we still have there a lot of outliers. The outliers will be analysed in next sections. Another variable, which needs to be transformed is 'neighbourhood cleansed'. It is a great indicator of listing position on the map. Moreover, the dataset contains all values and there are no missing ones. As there are 254 neighbourhoods, it could simply not give a good estimate in the prediction model due to lack of enough data

points in each of the neighbourhoods and the complexity of model would increase as it is a categorical variable and we would need to treat it by one hot-encoding method. Therefore, for further analysis and work we need to transform this feature. Reduce number of neighbourhoods and cluster them based on correlating factors is the objective of further analysis.

From the previous theoretical work, we have discussed that for feature reduction one of the powerful methods is PCA. As the first step we had to do the transform neighbourhoods, as categorical variable, to numerical variable. As we discussed before for this we are going to utilize one-hot encoding. Following analysis shows results:

```
PCA_LA_data = LA_DATA_v2[['neighbourhood_cleansed', 'price']]
PCA_LA_data = pd.concat([PCA_LA_data, pd.get_dummies(PCA_LA_data['neighbourhood_cleansed'],
                                                     prefix='neighbourhood_cleansed')],axis=1)
PCA_LA_data.drop(['neighbourhood_cleansed'],axis=1, inplace=True)
```

Figure 36 PCA

We will work also with a price as we believe there should be a relationship between location and price. The one-hot encoding is responsible for transformation to features values and assigning 1 to each record for which it is relevant. After that we have conducted PCA analysis:

```
pca = PCA(1)
pca_analysed = pca.fit_transform(PCA_LA_data)
print(PCA_LA_data.shape)
print(pca_analysed.shape)
```

```
(31253, 255)
(31253, 1)
```

Figure 37 PCA Results

As we can see we have reduced number of features from 255 to only one, in PCA called component with following values:

```
pca_analysed
```

```
array([[ 2819.90595361],
       [ -130.0940376 ],
       [ -125.09403762],
       ...,
       [  319.90601372],
       [ -115.09403798],
       [  319.90601372]])
```

Figure 38 PCA Array

When we analysed correlation between PCA values and the rest numerical values, we have find out that the relationship is not significant, and the analysis didn't bring expected results. As we still consider geolocation as a possible value predictor, we had try to use another two clustering methods: K-means clustering and Density-Based Spatial Clustering of

Applications with Noise or shortly DBSCAN. All these are part of sklearn library for python. K-means is computing distance between the point and each cluster center. DBSCAN also groups the closest data points together, based on distance, usually Euclidean and minimum number of points to be grouped. Advantage of this algorithm is that it also points out outliers or data points lying too far to be clustered [103].

With the K-means we were able to cluster them but without any specific effect and relationship to other variables as well as to the dependent one. DBSCAN gave us first 182 clusters, which is way too much. With the different parameters we were able to reduce them, but the results were not as we expected:

```
from sklearn.cluster import DBSCAN
from sklearn import metrics
from sklearn.preprocessing import StandardScaler
```

```
db = DBSCAN(eps=0.5, min_samples=10).fit(DBSCAN_LA_data)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_clusters_
```

```
182
```

Figure 39 DBSCAN Implementation

In the end we have decided to group the neighbourhoods based on their districts or regions. For this, we had to use external data sources as the datasets didn't come with districts. We pair our neighbourhoods with the districts from the LA Times Map directory [104]. We were able to match all of our neighbourhoods with the provided regions as follows:

```
def convert_to_regions(val):
    return dict[val]
```

```
LA_DATA_v2['regions'] = LA_DATA_v2['neighbourhood_cleansed'].apply(convert_to_regions)
```

```
LA_DATA_v2['regions'].describe()
```

```
count            31253
unique              16
top       Central L.A.
freq             11253
Name: regions, dtype: object
```

Figure 40 New Variable Derivation - Regions

And following are the new regions, which were mapped to the existing neighborhoods:

Antelope Valley	San Gabriel Valley	San Fernando Valley	Harbor
South L.A.	South Bay	Central L.A.	Westside
Santa Monica Mountains	Verdugos	Southeast	Eastside
Northwest County	Angeles Forest	Northeast L.A.	Pomona Valley

Table 1 Los Angeles Regions

4.4.4 Multivariate Analysis

In following section, we will examine relationships between individual features and their impact on target variable of listing price. Now we are going to look at the bivariate relationship between numerical variables and our target variables of sales.

```
features = pd.melt(LA_DATA_v2, id_vars=['price'], value_vars=sorted(num_f))
grid = sns.FacetGrid(features, col='variable', col_wrap=4, sharex=False, sharey=False)
plt.xticks(rotation='vertical')
grid = grid.map(sns.regplot, 'value', 'price', scatter_kws={'alpha':0.3})
[plt.setp(ax.get_xticklabels(), rotation=60) for ax in grid.axes.flat]
grid.fig.tight_layout()
plt.show()
```

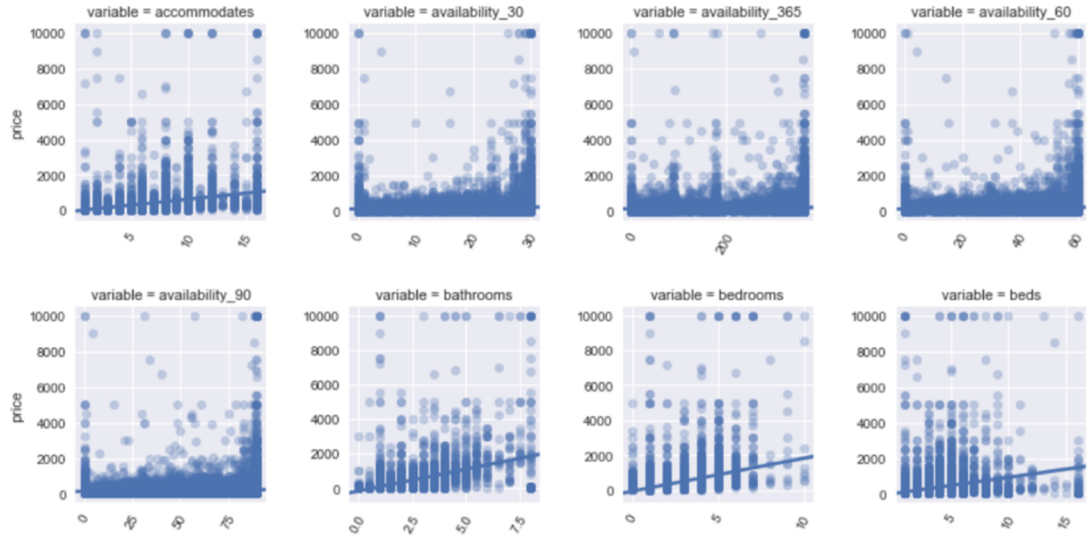


Figure 41 Bivariate Analysis 1

The first part of bivariate analysis showed us that not all variables have correlation with our target variable. Looking at availability variables gives us impression that there is no correlation at all. On the other side there seems to be a relationship between price and accommodates, bathrooms, bedrooms and beds.

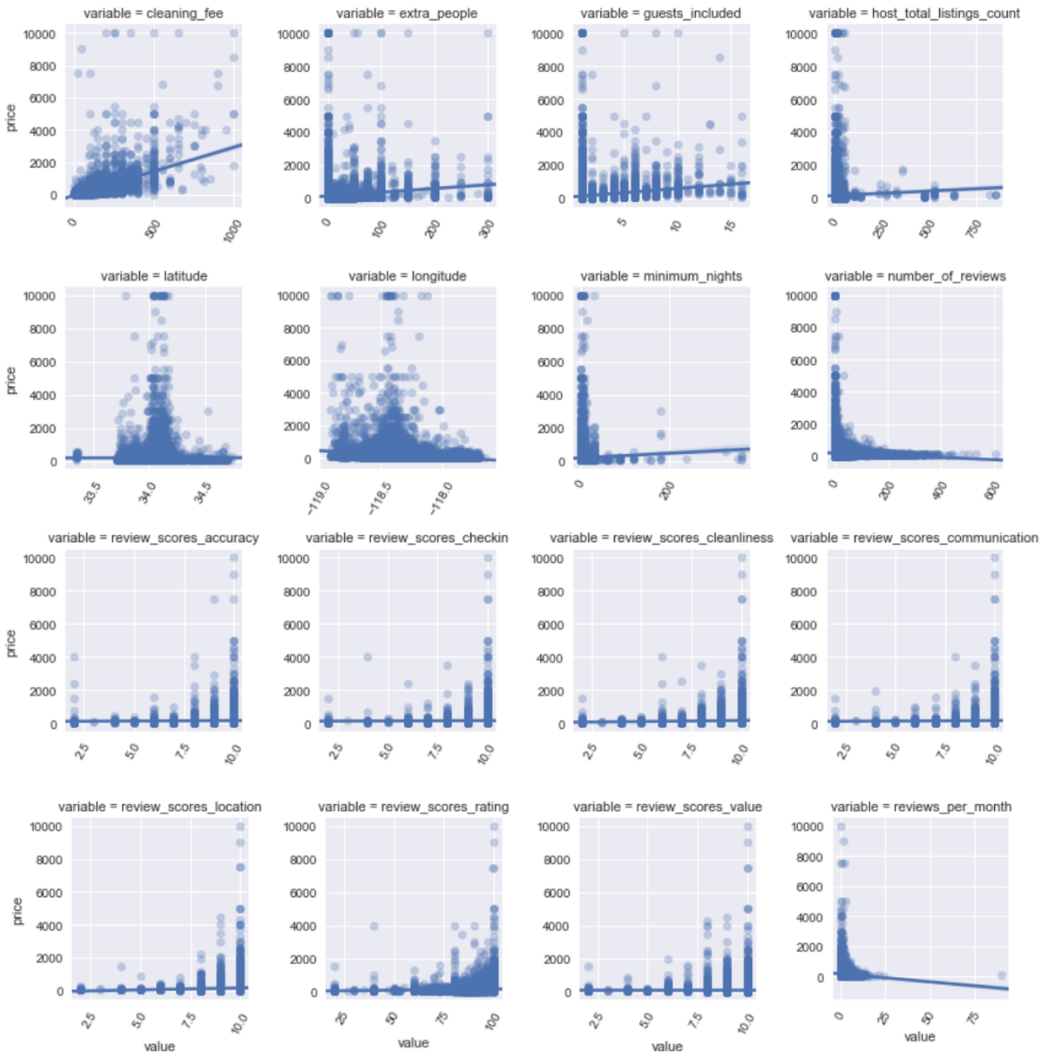


Figure 42 Bivariate Analysis 2

There are only few variables, with the impact on the target variable: ‘cleaning_fee’, ‘extra_people’, ‘guests_included’and ‘hosts_total_listings_count’. It is interesting finding that review scores are not very significant for price as we thought they would be. When we are comparing the variables with the log transformation of the price, we can observe that correlation becomes more linear:

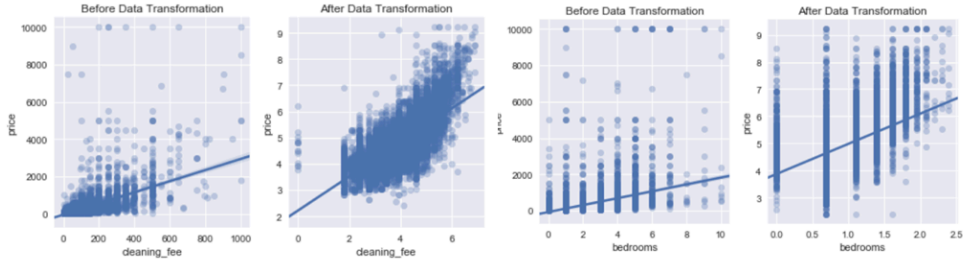


Figure 43 Bivariate Analysis 3

From the multivariate point of view, it is also interesting to see what relationships are between other variables. As we described in theoretical part, some algorithms exclude features with low or zero coefficients, others are keeping them in case they have correlation with the features with stronger connection to dependent variable. Following correlation matrix shows all numerical features and their correlations:

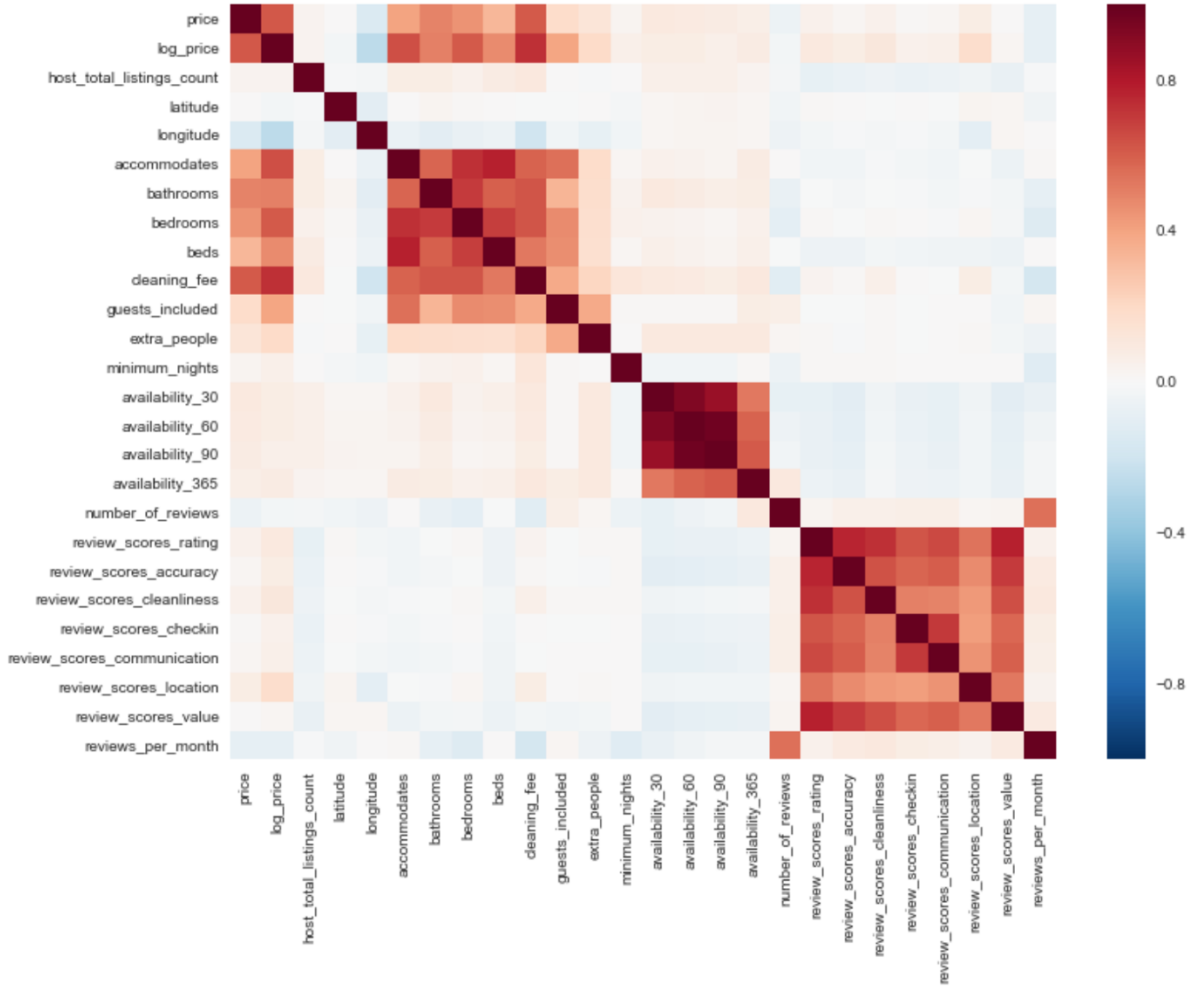


Figure 44 Correlation Matrix

Based on the results we can say that there are few variables with strong correlation to the price. As we can see, when transforming price variable, the correlation between logarithmic version of price is in most cases higher. Additionally, there are groups of variables, where the relationship is stronger. Availability variables have very strong correlation between each other as well as review score variables. Interesting observation is at the variable 'review_scores_location' and 'log price'. Even though the relationship is not strong, we can assume that location also plays a role as we assumed. There are also logical findings as correlation between accommodates and number of beds or beds and bedrooms. To sum up the topic of numerical variables and their correlations, following analysis sorts the most important features to 'price' as well as 'log price' variables.

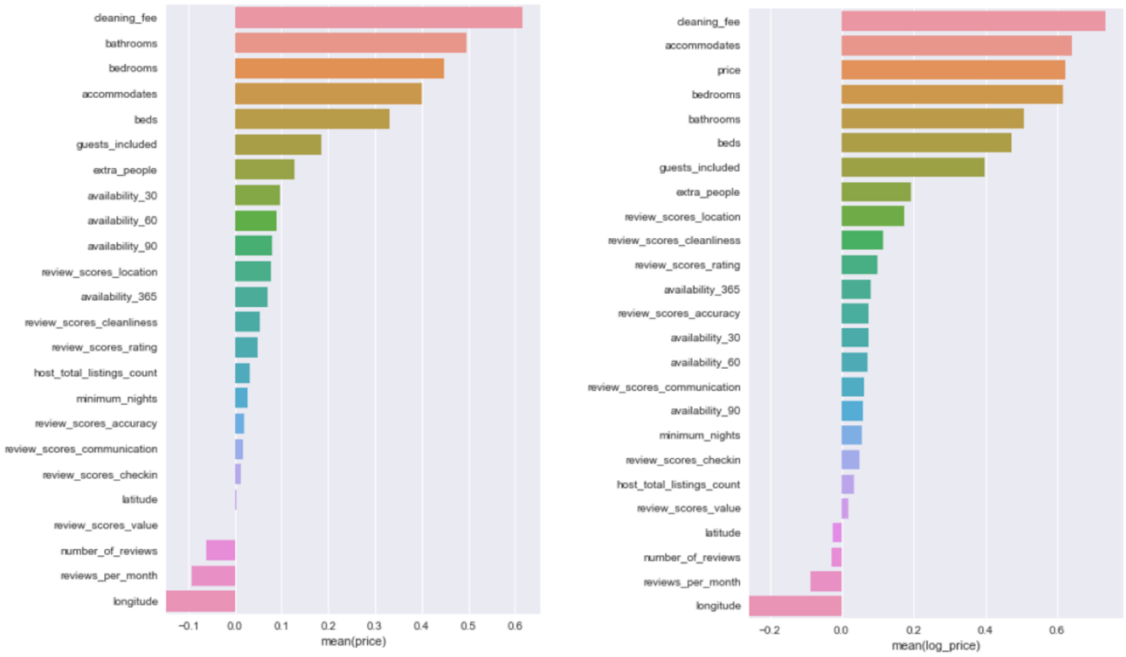
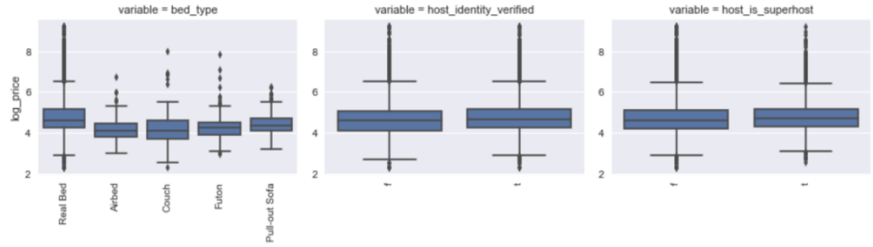


Figure 45 Transformed vs. Untransformed Price Correlation

Surprisingly but logically there is strong correlation of price and cleaning fee. The cleaning fee is not directly a part of the price for room. But it is logical that the final price consists from rent price as well as cleaning fee. The most correlated variables are also bathrooms, bedrooms, accommodates, followed by beds and guests included. As we also point out several times, log transformation of price shows stronger correlation to most of the numerical variables. We need to say that there are still outliers which might mislead results of analysis. We are going to deal with them in further parts of the research.

Next in the process pipeline is to understand the how categorical data are affecting price and what relationship they could possibly have. We will work with ‘log_price’ for this analysis as the previous work showed us better results after log transformation and more linear relationships.

```
# Count plots of categorical features
features = pd.melt(LA_DATA_v2, id_vars=['log_price'], value_vars=sorted(cat_f))
grid = sns.FacetGrid(features, col='variable', col_wrap=3, sharex=False, sharey=False, size=4)
grid = grid.map(sns.boxplot, 'value', 'log_price')
[plt.setp(ax.get_xticklabels(), rotation=90) for ax in grid.axes.flat]
grid.fig.tight_layout()
plt.show()
```



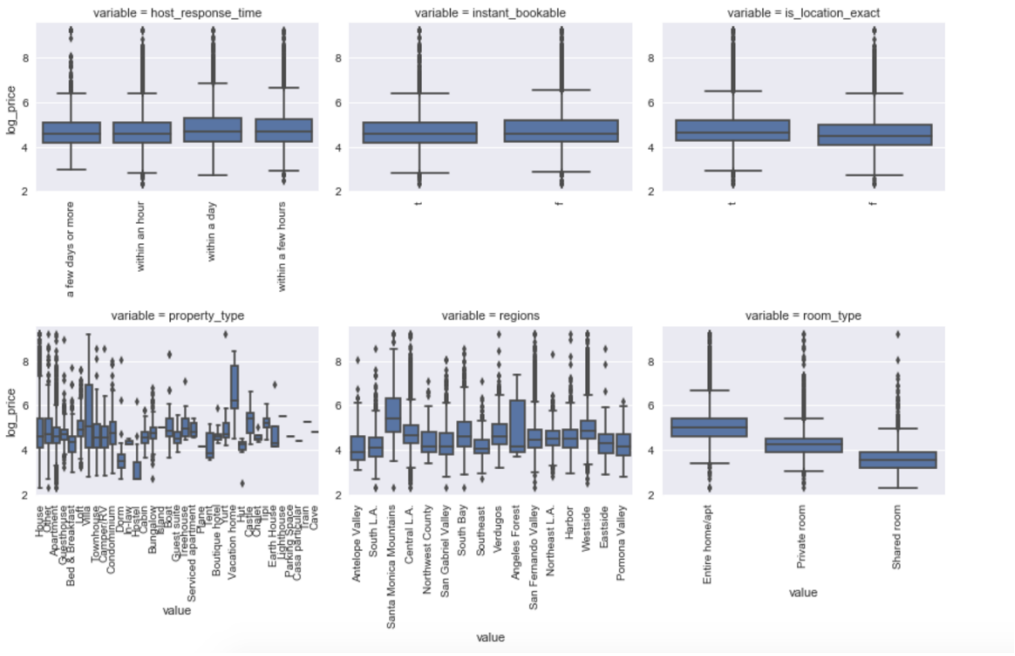


Figure 46 Categorical Features Bivariate Analysis

As observe from the analysis results, there are some which are showing significant variance in the mean o log price. As we expected, we have good results for regions, property_type or room_type. Our transformation has confirmed that between location and price there is correlation. To better understand which of categorical variables are significant we conduct one-way ANOVA to see the impact of each categorical feature.

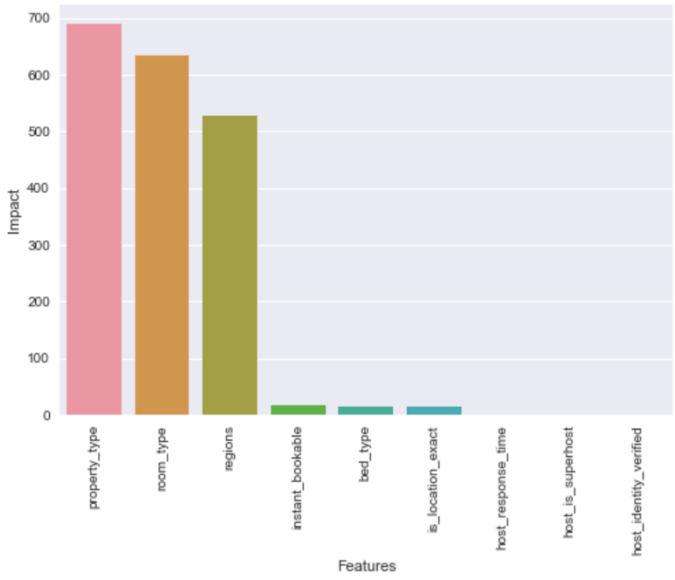


Figure 47 Categorical Features ANOVA

As we supposed, the three mentioned variables have shown us expected results. The strong correlation is between price and regions, property_type and room_type.

4.4.5 External Feature Integration

As we discussed before, sometimes the datasets don't come in the best shape, form or with what we exactly need. Most of the times it is necessary to conduct one or more feature engineering methods. We were already transforming some variables, now the result of this section is integration of open source datasets, which we believe might support performance of prediction models. Open data sites are getting more and more common. Even some of them make the data accessible via APIs, which is a huge help. Especially governments are making a lot of data available online about public with the aim to increase participation of citizens, to develop opportunities for business and to establish new communication channel with public. For our case there are relevant data from US environment, specially Los Angeles county. There are many websites, fulfilling our objective, we worked with following:

- data.gov - The hub of the U.S. Government's open data
- huduser.gov - The U.S. Department of Housing and Urban Development's data
- data.lacity.org - Information, Insights, and Analysis from the City of Los Angeles
- ffiec.gov - The Federal Financial Institutions Examination Council

For the data acquiring we applied web scrapping methods, APIs and downloads. For data manipulation excel, Jupyter notebook and pandas python package. Examining all kinds of possible datasets which would be interesting for prediction model and possible to integrate we have chosen following ones:

- US Census Data by County[2] – the whole website is aimed to provide datasets about US population from different point of views, considering demographics, income, population and housing data summary.
- Active Restaurants Heat Map Data[3] – containing information about active restaurants in the area.

US Census Data by county are provided by The Federal Financial Institutions Examination Council, possible to download manually – by state, county and tract code. A census tract code is a numeric specification of location, which is approximately equivalent to neighbourhood. In comparison with districts, tract codes are covering smaller areas but bigger that blocks. The main objective of census tracts is to give a small stable set of units for location identification [105]. The reason why not to use zip codes instead of tracts is that zipcodes of areas might change but tracts stay the same. We will use tracts to merge our dataset with the external.

[2] https://www.ffiec.gov/census/default.aspx

[3] https://data.lacity.org/A-Prosperous-City/Active-Restaurants-Heat-Map/gtcn-kik7

As mentioned before the datasets from the website might be acquired by two ways – download manually to pdf and transform or use web scrapping. Unfortunately, ffiec.gov doesn't provide API, which would make our work easier. Web scrapping is crucial for big data obtaining where we extract information from online sources [106]. As downloading would be time consuming, we decided to automatically pull data with web scrapping. In the url there is a following pater:

https://www.ffiec.gov/census/report.aspx?year=2017&county=037&tract=ALL&state=06&report=demographic

Figure 48 Web Scrapping URL

Year of download, first number – 037 - represents country, in this case California. 06 is the county reference – in our case Los Angeles County. Last part is name of report. We will be iterating process for: Income, Demographic, Housing and Population. As the result of the data integration from ffiec.gov we have successfully added following 26 variables:

```
In [138]: LA_DATA_merged.columns

Out[138]: Index(['id', 'Total Housing Units', '1- to 4- Family Units',
                 'Median House Age (Years)', 'Inside Principal City?',
                 'Owner Occupied Units', 'Tract Income Level',
                 'Distressed or Underserved Tract', 'Tract Median Family Income\n%',
                 '2017 FFIEC Est.MSA/MD non- MSA/MD\nMedian Family Income',
                 '2017 Est.\nTract Median Family Income',
                 '2015 Tract Median Family Income', 'Tract Population',
                 'Tract Minority %', 'Minority Population', 'Vacant Units',
                 'Owner Occupied 1- to\n4- Family Units', 'Renter Occupied Units',
                 '2015 Tract Median Household Income', 'Number of Families',
                 'Number of Households', 'Non-Hisp White Population',
                 'American Indian Population',
                 'Asian/ Hawaiian/ Pacific Islander Population', 'Black Population',
                 'Hispanic Population', 'Other Population/ Two or More Races'],
                dtype='object')
```

Figure 49 Integrated Variables

The integrated data contains again mixed datatypes of categorical, Boolean as well as numerical data types. The new features might potentially increase performance of our prediction model. Number of households, Median Income of Household, Renters Occupied Units, Owners Occupied Units or different kinds of population might also bring insights what kind of group of people are having most listings and how it impacts the price.

The biggest challenge, when integrating external data sources is to determine the best way how to map them into our case. Finding a proper key of mapping the datasets is time-consuming, but hopefully rewarding. In our case we have following variables, which might be used for datasets merging:

longitude/latitude	Exact location of listing based on coordinates
neighbourhood_cleansed	Exact neighborhood name of listing
zipcode	Location identification system used by postal services

Table 2 Possible mapping variables to external sources

To merge tract codes together with zip codes the mapping is required. This is provided by huduser.gov[4] where we have files that allocate ZIP codes to Census tracts. Tract code is consisting of 3 parts – state code, county code and tract code itself. Following is the example of mapping:

Zip code	Tract Code
90027	06037190401

Table 3 Tract Code Mapping Example

The second source of data is coming from LA open data, where we can find many interesting datasets. We have decided to integrate count of restaurants per zip code. The dataset is based on "active" businesses in the area of interest, which is updated monthly [107]. The dataset is integrated this time based on zipcode. The data manipulation was again required as we had each restaurant per zip. We have calculated the count of restaurants per zip. After data manipulation and merging we have successfully integrated new variable 'LA_restaurants_frequency' per zip code.

```
LA_restaurants_frequency = LA_restaurants.groupby('ZIP Code').size()
```

```
LA_DATA_merged3 = pd.merge(LA_DATA_merged2, LA_restaurants_frequency, how='left', on=['zipcode'
```

```
LA_restaurants_frequency.head()
```

	zipcode	Number of Restaurants per Zip
0	90028	208
1	90012	172
2	90006	159
3	90020	138
4	90036	137

Figure 50 Restaurants per Zip

We believe that open source data might bring unexpected insights into any predictive modelling. The advantage is that they are easily accessible and free.

[4] https://www.huduser.gov/portal/datasets/usps_crosswalk.html

4.4.6 Additional Feature Engineering

There are still variables which require additional feature engineering. The reasons are different. Starting with ‘review’ variables, which are representing basically the same values, but from different point of views. We will transform these features into single one. The more significant reason is multicollinearity, which we should aim to avoid. Usually in every regression problems we have some rate of multicollinearity, but in case of reviews, we can avoid it as the variables are representing similar output, from 1-10. As we could see, all ‘review’ variables are in strong correlation with each other. Here is the transformation of review variables:

```
LA_DATA_v6['review_overal_score'] = LA_DATA_v6[["review_scores_value", "review_scores_location",
                                                "review_scores_checkin", "review_scores_cleanliness",
                                                "review_scores_accuracy", "review_scores_communication"]].mean(axis=1)
```

Figure 51 Review Variables Trasformation

As the weight of all features is the same, we have decided to conduct mean operation to get overall review score. A similar situation applies when we consider ‘availability’ variables, where we have strong correlation between those 4 variables: ‘availability_365’, ‘availability_90’, ‘availability_60’, ‘availability_30’. We have created new variable instead, called ‘availability rate’. The result of this variable transformation is new variable ‘availability_rate’, with values between 0-1.:

```
LA_DATA_v6['availability_rate'] = (LA_DATA_v6["availability_365"]/365 + LA_DATA_v6["availability_90"]/90 +
                                   LA_DATA_v6["availability_60"]/60 + LA_DATA_v6["availability_30"]/30 )/4
```

```
LA_DATA_v6 = LA_DATA_v6.drop(["availability_365", "availability_90",
                              "availability_60", "availability_30"], axis=1)
```

Figure 52 Availability Variables Transformation

Next, we transform newly integrated categorical variable: Tract Income Level, consisting of 4: Low, Moderate, Medium, Upper, plus Unknow which we will treat as missing value:

```
LA_DATA_v8 = LA_DATA_v8.replace({"Tract Income Level" : {"Low" : 1, "Moderate" : 2,
                                                         "Middle" : 3, "Upper" : 4}})
```

Figure 53 Tract Income Variable Transformation

To finalize transformation of variables, we also need to change a representation of host_response_rate from string percentage to integer or float:

```
LA_DATA_v12['host_response_rate'] = LA_DATA_v12['host_response_rate'].str.rstrip('%').astype('float') / 100.0
```

Figure 54 Host Response Rate Variable Transformation

4.4.7 Missing Values Analysis

As our dataset still contains a lot of missing values, we need to conduct several operations to either impute values or drop them. Following figure shows percentage of missing values in our feature reduced dataset:

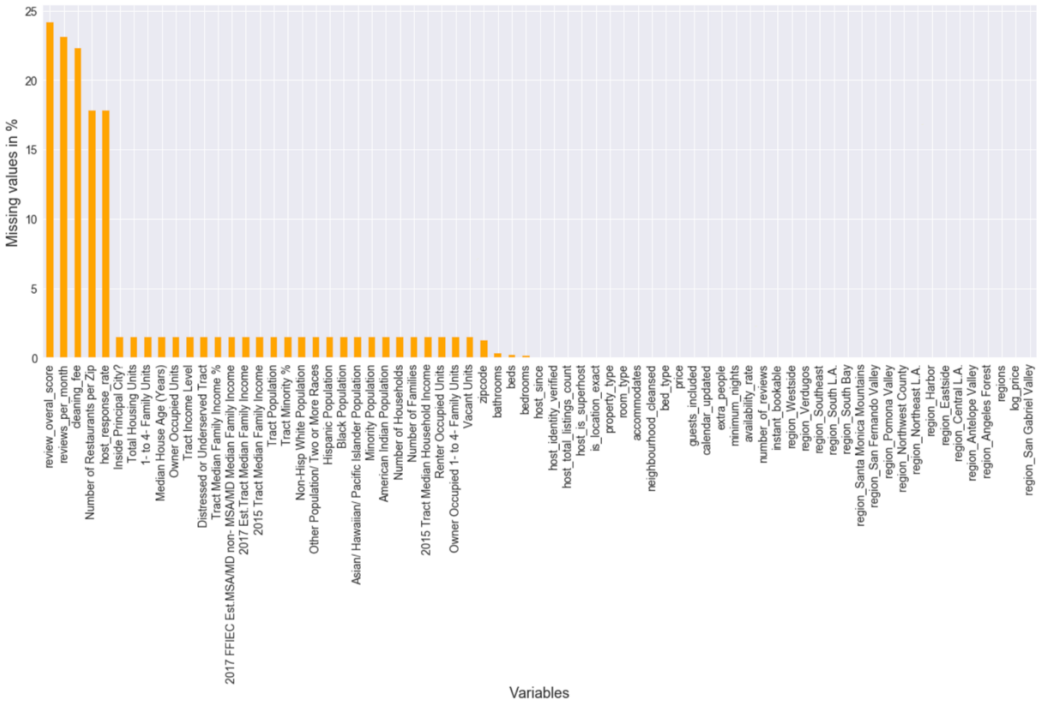


Figure 55 Missing Values analysis 1

After closer examination there are still variables with high rate of missing values. Referring to ‘review_rate’, ‘reviews_per_month’ or cleaning_fee which are reaching almost 25% of all missing values. Another group of variables is closely following them: ‘Number of Restaurants per Zip’ and ‘host_response_rate’. The third one is very interesting. It is related to missing zip code. As we integrated new features based on zip variable, we have almost the same rate of missing values for all of them as for zip. We started with zip codes, which might solve overall problem of missing values. As we know, zip codes are closely but not directly related to neighbourhoods. Following analysis shows count of total missing zip codes and unique neighbourhoods:

```
LA_DATA_v6_missing_zip = LA_DATA_v6[LA_DATA_v6['zipcode'].isnull()]
```

```
LA_DATA_v6_missing_zip['neighbourhood_cleansed'].describe()
```

```
count           395
unique          112
top       Hollywood
freq             22
Name: neighbourhood_cleansed, dtype: object
```

Figure 56 Zip Code Missing Values Analysis

Based on above results we see that there are 395 missing zip codes with unique 112 neighbourhoods. Our aim is to impute missing zip code-based frequency per neighbourhood. According following steps, we have successfully filled missing values for zip codes and integrated external data:

1. Find most frequent zip code per Neighbourhood – most of the neighbourhoods had usually one zip code, in other cases more zip codes with one extremely prevailing
2. Next, we imputed missing zip codes
3. Lastly, we mapped and imputed rest of zip code relevant data

We have successfully mapped all remaining zip codes, but unfortunately not all zip code relevant variables have been imputed. There is still missing approximately 120 missing values, which is less than 0.5%. We will continue in the process of data imputation of mentioned variables by mean and mode. Almost all of the variables are numerical, continuous except 'Tract_Income_Level'. As the rate of the missing values is low, this will be sufficient procedure for all zip code relevant variables plus number of bedroom, beds and bathrooms. Firstly, by a mode and by mean:

```
cols_mode = ["beds", "bedrooms", "bathrooms", "Tract_Income_Level", "Inside Principal City?"]
```

```
mode = LA_DATA_v8.filter(cols_mode).mode()
print (mode)
```

```
   beds  bedrooms  bathrooms  Tract_Income_Level Inside Principal City?
0   1.0       1.0        1.0                 4.0                    Yes
```

```
LA_DATA_v8[cols_mode]=LA_DATA_v8[cols_mode].fillna(LA_DATA_v8.mode().iloc[0])
```

```
cols_mean = ['Total Housing Units','1- to 4- Family Units', 'Median House Age (Years)',
             'Inside Principal City?', 'Owner Occupied Units', 'Distressed or Underserved Tract',
             'Tract Median Family Income %','2017 FFIEC Est.MSA/MD non- MSA/MD Median Family Income',
             '2017 Est.Tract Median Family Income','2015 Tract Median Family Income', 'Tract Population',
             'Tract Minority %', 'Minority Population', 'Vacant Units','Owner Occupied 1- to 4- Family Units',
             'Renter Occupied Units','2015 Tract Median Household Income', 'Number of Families',
             'Number of Households', 'Non-Hisp White Population', 'American Indian Population',
             'Asian/ Hawaiian/ Pacific Islander Population', 'Black Population','Hispanic Population',
             'Other Population/ Two or More Races']
```

```
mean = LA_DATA_v8.filter(cols_mean).mean()
print (mean)
```

```
Total Housing Units                                  1809.472808
1- to 4- Family Units                                 865.168546
Median House Age (Years)                               53.102663
Owner Occupied Units                                  565.373775
Tract Median Family Income %                          131.438913
2017 FFIEC Est.MSA/MD non- MSA/MD Median Family Income  64287.607208
2017 Est.Tract Median Family Income                   84515.265652
2015 Tract Median Family Income                       82419.180013
Tract Population                                     3942.950371
Tract Minority %                                       52.611964
Minority Population                                  2064.554110
Vacant Units                                          146.108991
Owner Occupied 1- to 4- Family Units                  468.305612
Renter Occupied Units                                1097.990042
2015 Tract Median Household Income                  67813.911567
Number of Families                                    807.467862
Number of Households                                 1663.358678
Non-Hisp White Population                            1878.396261
American Indian Population                             13.167903
Asian/ Hawaiian/ Pacific Islander Population          597.677942
Black Population                                      234.738267
Hispanic Population                                  1084.840031
Other Population/ Two or More Races                   134.129967
dtype: float64
```

```
LA_DATA_v8[cols_mean]=LA_DATA_v8[cols_mean].fillna(LA_DATA_v8.mean().iloc[0])
```

Figure 57 Missing Values Handling

In the next part of missing values, we will be dealing with the ones where the rate is much higher in comparison with previous ones. We are referring to following features: 'review_overal_score', 'reviews_pre_month', 'Number of Restaurants per Zip', 'cleaning_fee' and 'host_response_rate'. For the mentioned ones, we will use more precise method of clustering - KNN. The advantage of KNN is that is looks for k neighbours and based on this value assign the missing one. We will apply 'fancyimpute' Python package for matrix completion and feature imputation algorithms as follows:

```
from fancyimpute import KNN
```

```
train_df_cols=list(KNN_LA_data)

source = pd.DataFrame(KNN(k=5).complete(KNN_LA_data2))

source.columns=train_df_cols
```

```
Imputing row 1/31253 with 0 missing, elapsed time: 712.882
Imputing row 101/31253 with 1 missing, elapsed time: 713.258
Imputing row 201/31253 with 1 missing, elapsed time: 713.459
Imputing row 301/31253 with 0 missing, elapsed time: 713.663
```

Figure 58 KNN Missing Data Imputation

To our expectations, the imputation was highly successful but required much more computational time and power that imputation methods before. Following figure shows us number of missing values after all imputation methods:

```
missing_values = sum(LA_DATA_v12.isnull().values.ravel())
print('Number of missing values in dataset is: ', missing_values )
```

```
Number of missing values in dataset is:  0
```

Figure 59 Missing Values analysis 2

Our dataset is now without any missing value and is prepared for next operations.

4.4.8 Outliers

We are still aware, the dataset definitely contains some outliers, responsible for strong right skewness. There are few listings with extreme values, affecting relationships between variables and target variable. Removing outliers is always a question and in larger datasets with higher dimensions it is difficult to detect them. In our case we will focus to remove outliers specially based on price. For our case we will utilize power of Interquartile Range Method.

As we don't want to remove sensitive data, we will remove outliers based on log transformation of price, which is already visibly removing impact of outliers in our dataset.

Usually to the value of IQR is 1.5 but it depends on some factors, which are affecting price. As we don't want to remove too much of data, we will consider value of 2 times of IQR for outlier removing and remove some of them also manually. Following analysis shows how to proceed with the outlier removal:

```
q75, q25 = np.percentile(LA_DATA_v13_out.log_price.dropna(), [75 ,25])
iqr = q75 - q25

min = q25 - (iqr*2)
max = q75 + (iqr*2)
```

```
i = 'log_price'

plt.figure(figsize=(10,8))
plt.subplot(211)
plt.xlim(LA_DATA_v13_out[i].min(), LA_DATA_v13_out[i].max()*1.1)
plt.axvline(x=min)
plt.axvline(x=max)

ax = LA_DATA_v13_out[i].plot(kind='kde')

plt.subplot(212)
plt.xlim(LA_DATA_v13_out[i].min(), LA_DATA_v13_out[i].max()*1.1)
sns.boxplot(x=LA_DATA_v13_out[i])
plt.axvline(x=min)
plt.axvline(x=max)
```

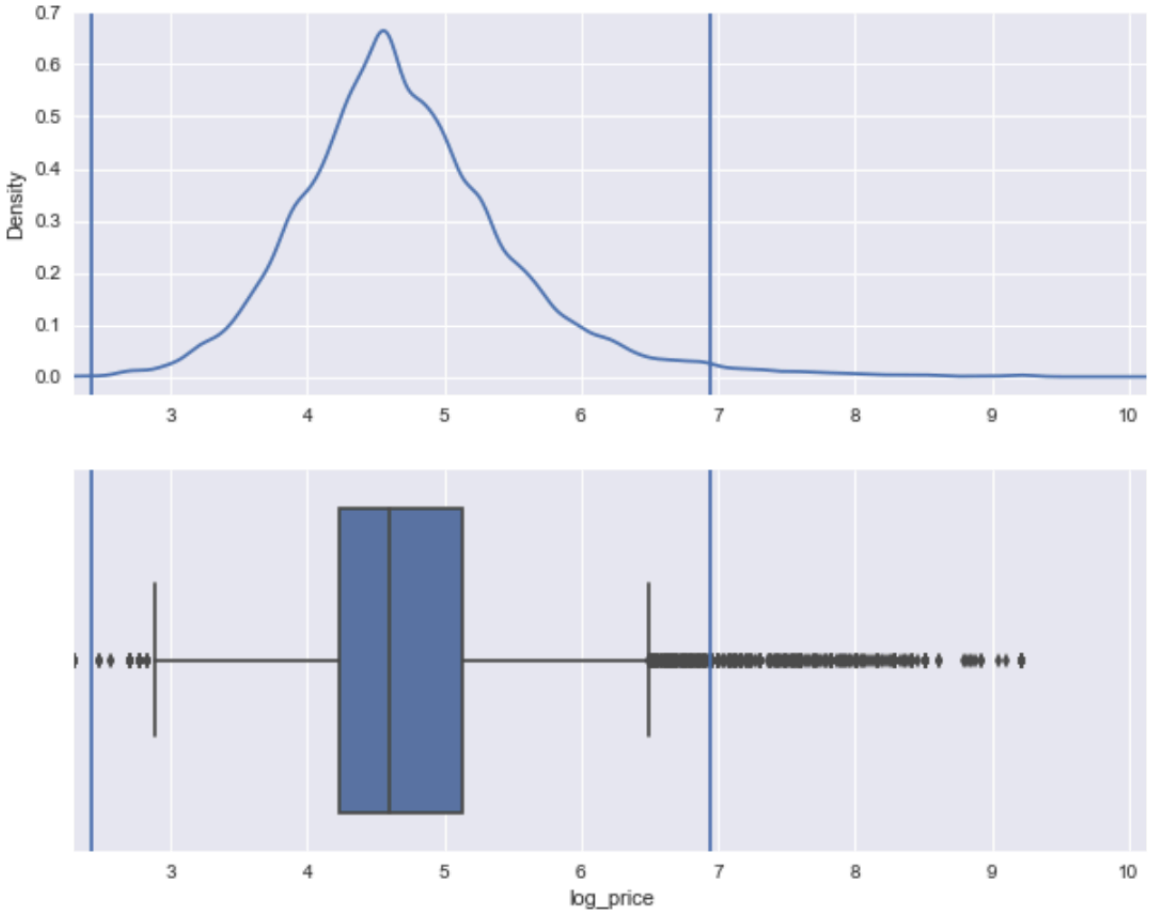


Figure 60 Outliers Analysis

As we can see only small portion of records are creating strong skewness off price. We observed in left true outlier representing very small prices for location as Los Angeles. There were shared dorm rooms, with extremely low rate. Same thing was on the right side. Extreme cases of unusual rentals for Airbnb. Moreover, there are also rare cases, usually isolated ones like in case of types of accommodation: train, cave, parking lot or boat. These outliers are removed manually, thanks to findings of multivariate analysis.

Starting with host_total_listing_count, where there is no listing present is simply misleading – we removed by this procedure 34 observations. Furthermore, when examining type of offered properties, there are many of very unusual type with low occurrence but rather high price: island, yurt, castle, lighthouse, cave or Earth house. We have dropped these, in total 30 records. With slightly higher frequency there are additional ones like hut, chalet, tent, boutique hotel and other, which considered to keep, because they don't represent extreme values and include them in category other.

4.4.9 Final Data Pre-processing

After all data preparation and feature engineering, the dataset is getting ready for model training and testing and evaluation. For better performance of the model there is still necessity to bring all variables into same range as well as encode the rest of categorical variables.

Final procedure requires to distinguish between categorical and numerical variables. As we are now aiming to transform highly skewed features, we will only consider the ones containing data types of integer or float. The log transformation of skewed variables. The procedure aims to adjust the data to be less skewed and better interpretable. After conducting log skewness analysis, we got following results:

```
skewness = LA_num.apply(lambda x: skew(x))
skewness = skewness[abs(skewness) > 0.5]
print("Number of skewed and transformed features :" + str(skewness.shape[0]))
skewed_features = skewness.index
LA_num[skewed_features] = np.log1p(LA_num[skewed_features])
```

```
Number of skewed and transformed features :38
```

Figure 61 Skewed Features Transformation

The procedure has transformed 38 skewed numerical features to make distribution more Gaussian.

Next step is to encode categorical data into language of machines – binary. There is a method called one-hot encoding, which creates translates categorical variables into 1s and

0s by transposing all values as features. In Pandas package, there is a function called get_dummies, which is equivalent to one hot encoding from sklearn library. Here is the result of variable encoding:

```
LA_cat = pd.get_dummies(LA_cat)
```

```
LA_cat
```

	host_is_superhost_f	host_is_superhost_t	host_identity_verified_f	host_identity_verified_t	is_location_exact_
0	1	0	0	1	0
1	1	0	0	1	0
2	1	0	1	0	0
3	1	0	1	0	0
4	1	0	0	1	0
5	1	0	0	1	0

Figure 62 Categorical Variables Encoding

Finally, we are able to split our dataset into training and testing. Models will be trained and then tested:

```
from sklearn.model_selection import cross_val_score, train_test_split
X_train, X_test, y_train, y_test = train_test_split(train, labels,
                                                    test_size = 0.3, random_state = 0)
print("X_train : " + str(X_train.shape))
print("X_test : " + str(X_test.shape))
print("y_train : " + str(y_train.shape))
print("y_test : " + str(y_test.shape))
```

```
X_train : (21499, 90)
X_test : (9214, 90)
y_train : (21499,)
y_test : (9214,)
```

Figure 63 Train/Test Data Split

Based on the theoretical knowledge from previous chapters, the feature scaling is required to prepare data for model training and fitting. The procedure which we apply is called standardization. With this method all numerical predictors are translated into scale with center of 0 and standard deviation of 1. By applying StandardScaler of sklearn library we obtained following results:

```
stdSc = StandardScaler()
X_train.loc[:, num_features] = stdSc.fit_transform(X_train.loc[:, num_features])
X_test.loc[:, num_features] = stdSc.transform(X_test.loc[:, num_features])
```

Figure 64 Standardization

To sum up the data pre-processing, we have conduct several steps towards successful model development. In next section we describe modelling phase.

4.5 Predictive Modelling

In this section we are finalizing the whole process of model development, where we train, validate, tune and test different models. Based on the results we select one with best performance. Putting all steps together should bring expected results of good performing model. Before we move any further we define our cross-validation function as follows:

```
n_folds = 5

def rmse(model, X, y):
    folds = KFold(n_folds, shuffle=True, random_state=42).get_n_splits(X.values)
    score= np.sqrt(-cross_val_score(model, X, y, scoring="neg_mean_squared_error", cv = kolds))
    return(score)
```

Figure 65 RMSE

The function splits the data into n number of folds and each of the folds is then used as a validation set once and the rest of data for training purposes.

4.5.1 OLS

As the first model we work with OLS model without any regularization. The model is simple, serves as a baseline model, which we iterate several times to calculate score. Following figure shows results:

```
Root Mean Squared Error on Training set for OLS : 0.35339161183896356
Root Mean Square Error on Test set set for OLS : 55490045.43770152
```

Figure 66 Residuals vs Predictions - OLS

As we can observe, model RMSE for the OLS is dealing with strong overfitting. The reason might be that we didn't conduct cross validation. The results will serve us to compare with the rest of the models. To conclude, the model residuals are randomly allocated around the line, which we consider as a good sign, meaning that most of the information were taken into account. But as we see the model didn't perform well. Tthe closer are training and testing RMSE results the better. There is deviation between two of them. OLS will serve as a comparison to the rest models.

4.5.2 Ridge regression

Next model, which we use too fit our data is Ridge regression which utilizes L2 norm, described in theoretical part. For our model implementation we are working with sklearn version Ridge Regression.

For tuning and searching of best hyper parameters of our model we apply Exhaustive Grid Search. Grid Search takes into account all parameter combinations and keeps the best one. The implementation is as follows:

```
from sklearn.linear_model import Ridge
from sklearn.model_selection import GridSearchCV

param_grid = {'alpha': [7., 7.5, 8, 8.5, 9, 9.5, 10., 10.5, 11, 11.5, 12., 12.5, 13.]}
ridge = GridSearchCV(Ridge(), cv=5, param_grid=param_grid, scoring='neg_mean_squared_error')
ridge.fit(X_train, y_train)
alpha = ridge.best_params_['alpha']
ridge = ridge.best_estimator_
```

Figure 67 Grid Search

We have established a grid for possible parameter values, and integrated functionality of the grid search gives us best recommended parameter. We iterate process several times to see, which hyperparameters are the best. After implementation, we have following results:

```
Root Mean Squared Error on Training set for RidgeCV : 0.3530608400755574
Root Mean Square Error on Test set set for RidgeCV : 0.3485138407937399
Best alpha parameter is around:  7.5
```

The results show us that training and testing RMSE have smaller difference between each other, meaning that we slightly improved the underfitting/overfitting issue from our baseline OLS model, which is good. Regularization created conditions for data to avoid overfitting. Looking at the prediction graphs we see that predicted values are centered around real values, which indicates a good model. Based on results we also see that testing data showed slightly better results:

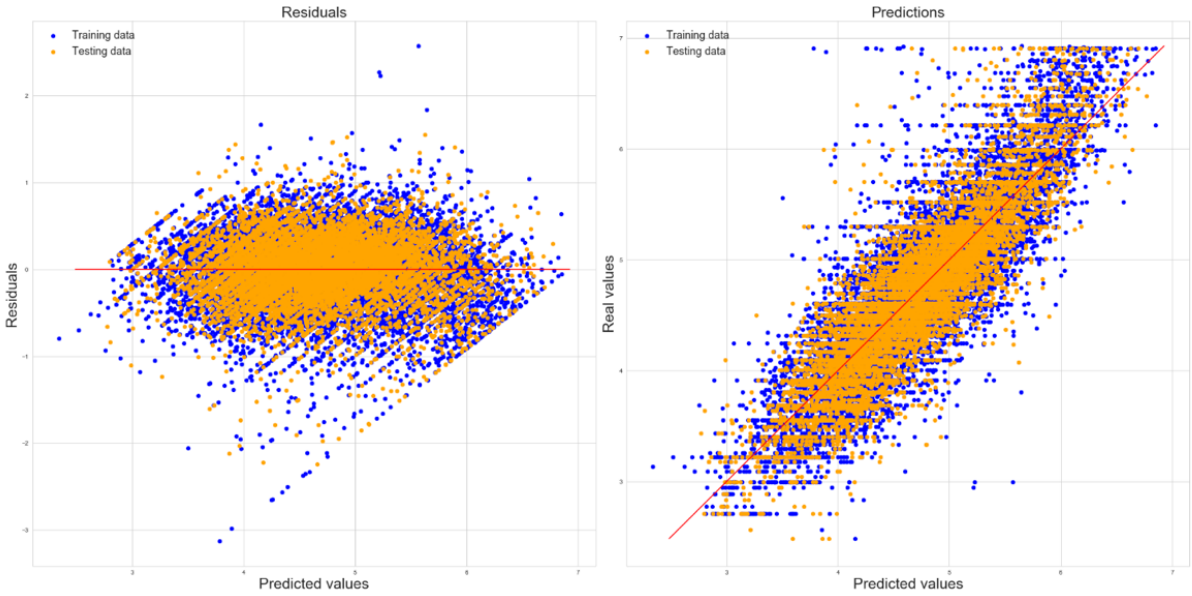


Figure 68 Residuals vs Predictions - Ridge

When examining the most significant features and their coefficient we work with following:

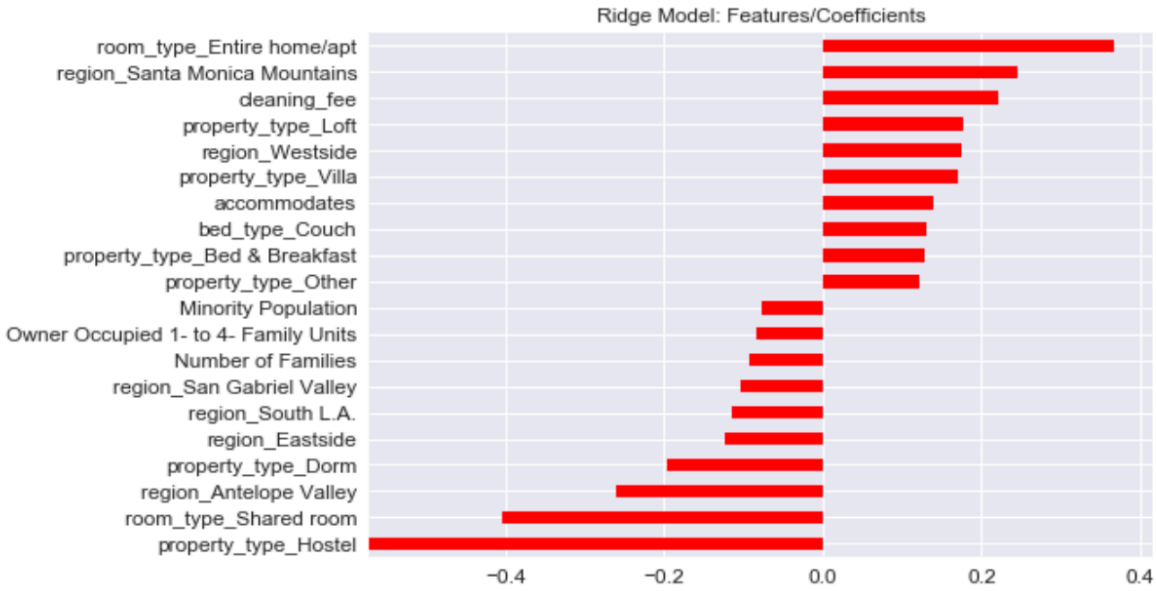


Figure 69 Feature Importance - Ridge

As we can see, Ridge model used all available features. As expected, the price is driven by type of room – the bigger place to stay, the higher price. Region plays also important role. More lucrative areas like LA Westside or Santa Monica Mountains increase price and vice versa. The price is lower in accommodation types of dorms or hostels. Strongly intriguing is to observe how new integrated data has been involved in prediction model. For negative coefficients we have Minority Population, Owner Occupied Units or Number of families. The algorithm also didn't drop any of the available features.

4.5.3 Lasso Regression

Similarly, with Ridge regression, we have Lasso regression, which is on the other side working with L1 norm. As already described, difference is in the way how algorithm takes into account features and assigned them coefficients. In case of Lasso, there might be coefficients of 0 as well, which means dropping. We also work with grid search in this case with following results:

```
Root Mean Squared Error on Training set for Lasso : 0.35299755782440323
Root Mean Square Error on Test set set for Lasso : 0.34843522665443827
Best alpha parameter is around:  0.0001
```

```
coefs = pd.Series(lasso.coef_, index = X_train.columns)
print("Lasso selected " + str(sum(coefs != 0)) + " features and dropped " +  \
      str(sum(coefs == 0)) + " features")
```

```
Lasso selected 76 features and dropped 14 features
```

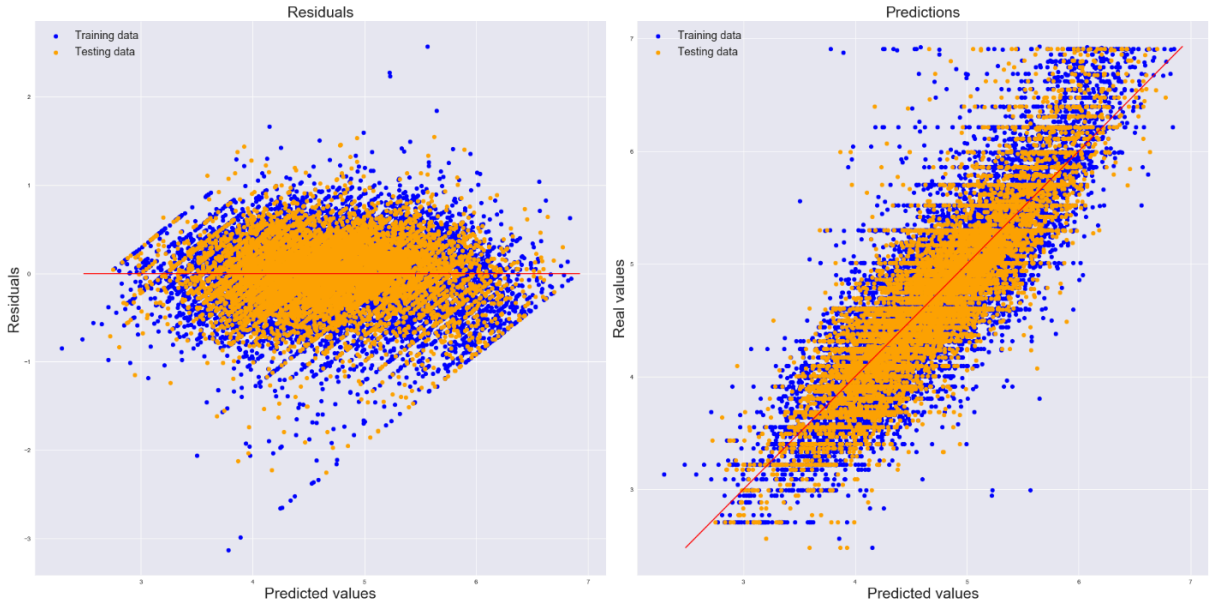


Figure 70 Residuals vs Predictions - Lasso

Despite the fact that Lasso was working with less features as we can see above, it has almost identical results with Ridge. Lasso dropped 14 features due to 0 coefficients assigned to them by algorithm itself. Examining the scatterplots, nothing deviates from results before. Looking at the most important features, the top 10 remained the same. But the difference is in the assigned coefficients to specific features. Predictor 'property_type_Hostel' received much higher negative coefficient as with Ridge, on the other side home/app has lower values as before:

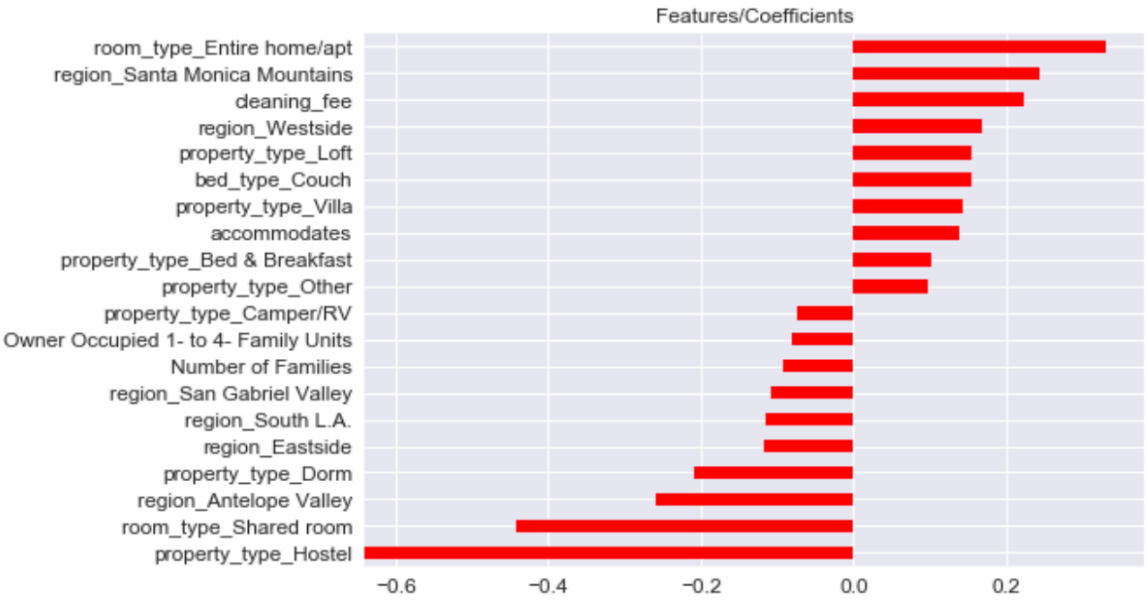


Figure 71 Feature Importance - Lasso

4.5.4 ElasticNet

ElasticNet considers both norms, L1 and L2. Nevertheless, when looking at the results, there is almost no difference in comparison with Lasso:

```
Best l1_ratio parameter is around  : 0.85
Best alpha parameter is around : 0.0001
Root Mean Squared Error on Training set for ElasticNet : 0.3530409361471627
Root Mean Squared Error on Training set for ElasticNet : 0.34843198992554536
ElasticNet selected 80 features and dropped 10 features
```

Having a look on the available results we can say that ElasticNet eliminated less features than Lasso, but the results are highly similar. Considering features as well as their coefficient, it is almost identical with Lasso.

4.5.5 Bayesian Ridge

We also implemented another regression algorithm on Bayesian approach to ridge regression. Nevertheless, the results are again similar to the others.

```
Root Mean Squared Error on Training set for Bayesian Ridge : 0.35840768973111836
Root Mean Squared Error on Test set for Bayesian Ridge : 0.3530666787839384
```

We have again applied grid search for model tuning as well as our cross validation. Importance of features stayed the same as with Ridge Regression.

4.5.6 XGBoosting

Based on first testing of XGBoosting we had to conduct feature selection due to the fact that it is much more computationally expensive and time-requiring. For dimensionality reduction we considered also PCA, but we had better results with XGBoost itself. The algorithm calculates feature importance scores which is used for features selection. After feature selection we got 41 features for further modelling.

As described before, XGBoosting is powerful model used for both regression and classification. It is based on gradient boosted decision trees for faster and better performance. We will again work with our rmse metric as well as grid search for best parameters. Here we can see RMSE results of XGBoosting:

```
Root Mean Squared Error on Training set for XGBoosting : 0.2993175125951638
Root Mean Squared Error on Test set for XGBoosting : 0.31538168769499375
```

As we can see, XGBoosting managed to have the lowest values of RMSE, which is a good indication. We further observe that there is slight deviation between train and test set.

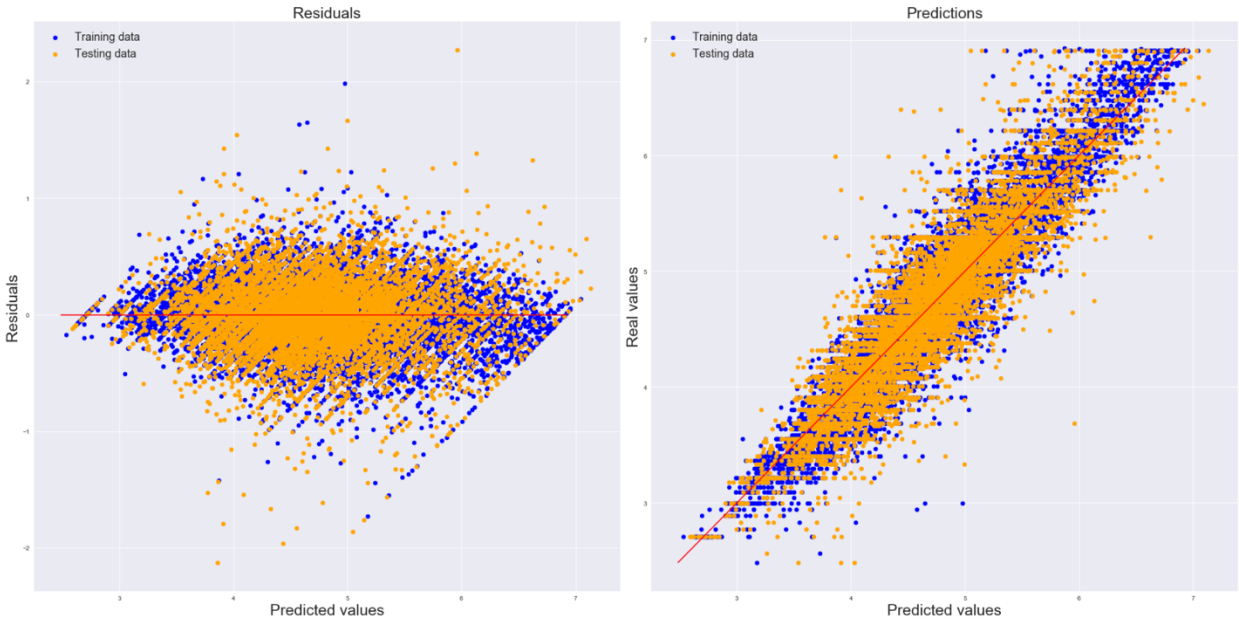


Figure 72 Residuals vs Predictions - XGBoosting

Based on the residual and prediction we can see that values are even more centered around the line. XGBoosting also considered different variables for the predictions: 'reviews_per_month', 'review_overall_score' or 'cleaning fee' are the most significant features. We can also notice that new integrated features played role in predictions like 'Number of Restaurants per Zip':

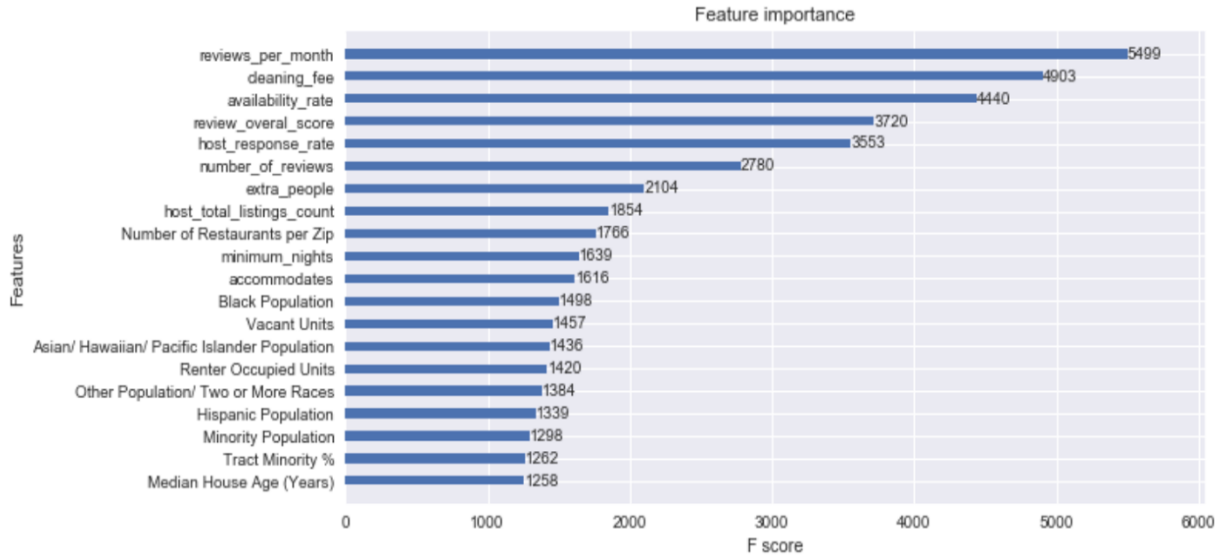


Figure 73 Feature Importance - XGBoosting

We can also notice that new integrated features played role in predictions like 'Number of Restaurants per Zip'. Following table shows tuned parameters:

Parameters	Best Value
colsample_bytree	0.4
gamma	0
learning_rate	0.07
max_depth	4
min_child_weight	1.5
n_estimators	1500
reg_alpha	0.75
reg_lambda	0.45
subsample	0.6
seed	42

Table 4 XGBoosting Tuned Parameters

To evaluate XGBoosting based on gathered results we can say that the algorithm has the best performance from the selected models, but it requires the most computational resources, especially for parameters tuning.

4.5.7 Support Vector Regressor

Last but not least we leverage power of support vector regressor. Applying the same selected features as in case of XGBoosting regressor with cross validation and grid search for parameters tuning, we observed much longer computational time than in case of first 5 linear regression models. Despite this fact we could observe increase performance and better results of SVR:

```
Root Mean Squared Error on Training set for SVR : 0.31334109913338215
Root Mean Squared Error on Test set for SVR : 0.324163743522127
```

As we can see, the values of RMSE with application of SVR is lower than in case of linear regression methods. Following figure shows residuals and predictions:

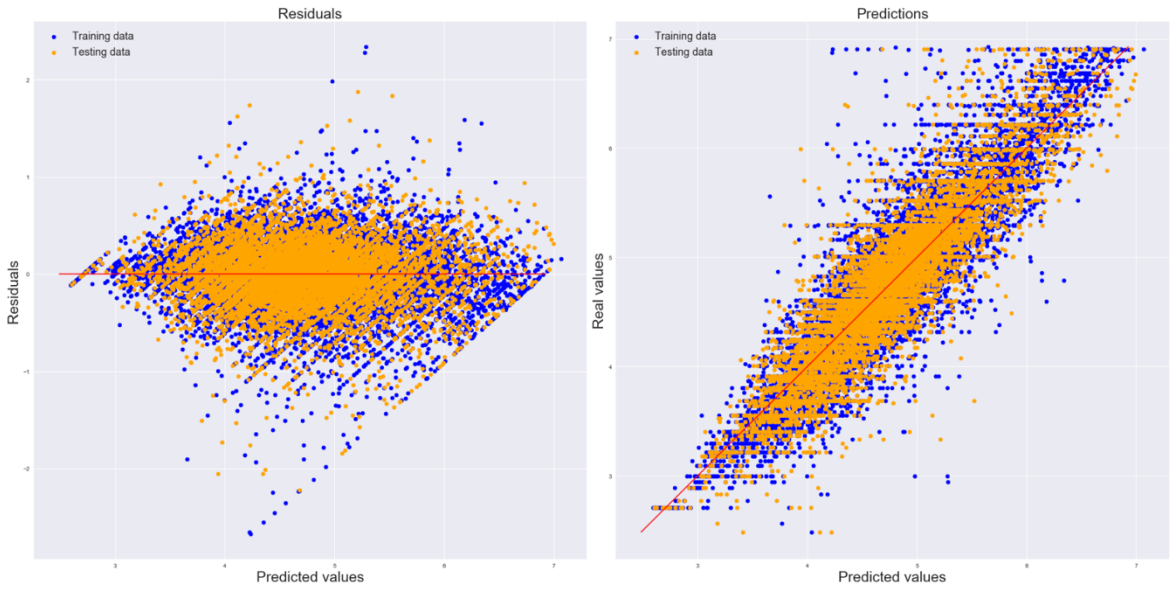


Figure 74 Residuals vs Predictions - SVR

We can observe that values are again closer. In general, SVR seems to be performing better than Linear Regression Models but worse than XGBoosting.

4.5.8 Evaluation

In our predictive modelling we have applied several machine learning models and algorithms to estimate price of Airbnb listings. Based on the RMSE metric, we obtained following results:

Model	Train Set Results(RMSE)	Test Set Results (RMSE)
OLS	0.3530	55490045.4377
Ridge	0.3530	0.3485
Lasso	0.3529	0.3484
ElasticNet	0.3530	0.3484
Bayesian Ridge	0.3584	0.3530
XGBoosting	0.2993	0.3153
SVR	0.3133	0.3241

Table 5 RMSE Results

As we can see, most of the linear regression models have performed quite well except the OLS, where we can see extreme results in case of test set. Two of the models, XGBoosting and SVR, performed the best based on gathered information. Next table compares results of median absolute error, reflecting price:

Model	Train Set Results (Median Absolute Error)	Test Set Results (Median Absolute Error)
OLS	0.2092	0.2105
Ridge	0.2083	0.2098
Lasso	0.2086	0.2086
ElasticNet	0.2086	0.2087
Bayesian Ridge	0.2098	0.2137
XGBoosting	0.1064	0.1626
SVR	0.1345	0.1714

Table 6 Median Absolute Error

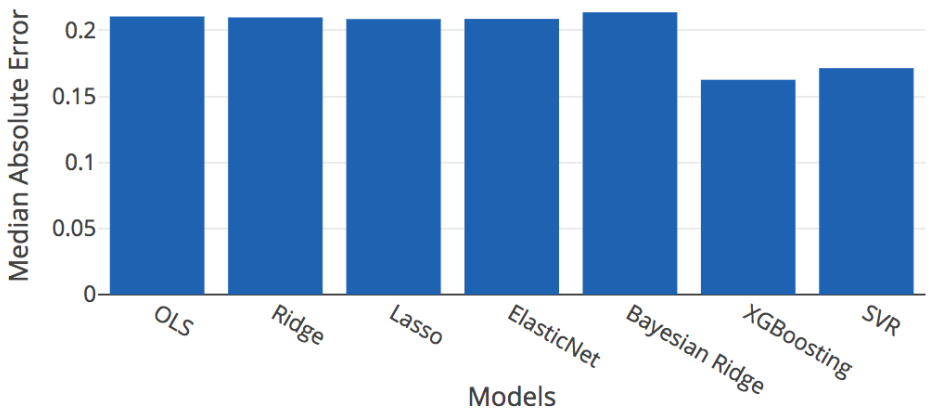


Figure 75 Median Absolute Error Models Evaluation

Based on the evaluation metrics of each model we can say that Extreme Boosting Regressor had the best results out of all implemented models. With the median absolute error of $16.26 we can say that the application of machine learning methods achieved expected goals and we developed well performing model. Comparing to other similar studies we have most precise results in regard to a real price of Airbnb listings.

We could observe interesting relationships between dependent variable and predictors of datasets. Out of all features, the most impactful factors influencing price are number of reviews per month and their score. As it is a common practice, people are making decision based on previous experience of the other guests and demand is forming price of the listing.

Cleaning fee is also one of the price influencing factors, which is obvious. With the amount of fee, it is expected that price for the whole listing will grow as well. Model was also improved by new integrated open source features like Number of Restaurants in the area. In general, the whole open source integration seems to have a positive effect on the price prediction and model performance. What especially took our attention was the impact of different social levels and structures of population. The shapes of districts characterized by majorities or minorities are in correlation with the price of Airbnbs. We believe that our model has brought different perspectives on what is responsible for price of Airbnb listings and helps either hosts or guest to estimate it more accurately.

5 Conclusion

The application of machine learning method in connection with housing price estimation proved to be very effective but requires a lot of resources and strong data-oriented approach. Without a proper and deep understanding of the data and its underlying structures it would not be possible to develop well performing model. As it was said many times, the data preparation and pre-processing themselves are the most important part when developing a prediction model. In our case the most important and demanding processes were feature engineering with external factor integration and missing value imputation. Except the predictive algorithms, data and its quality are playing the key role. When evaluating applied methods, we worked with multiple different approaches for each step and selected the best performing one. Every single one of them required a lot of preparation and deep understanding of background and its correct application.

Referring to the prediction results itself we can conclude that the performance of the models was good, especially in case of XGBooting. It proved that the correct application of chosen methods might bring expected results. With the median absolute error of $16.26 it is great results in comparison with similar studies. Surprising findings were related to predicting features. Some of them were expected but some of them very new. Speaking about the impact of the local services, structure of population or other social factors on Airbnb listing price was intriguing observation. In general, we are satisfied with our results.

The research brought us a lot of insights on how price prediction with machine learning works, what are the potentials as well as its limitations. Even the results achieved our initial objectives of the research there are still many possibilities how it might continue further. Next steps of the research might lead to integration of more predictors based on geo location or other features. As we have received a proof of the open source data power in our model, further research in this area might be conducted.

From the data scale point of view, next research might continue on the bigger rage, such as countrywide or even worldwide perspective of the data. Taking into account the presence of Airbnb in the most countries around the world, prices prediction might be examined also on this level.

With the further application of web scrapping methods, we can also consider a research in the area of text mining as well as object recognition. People's decisions are also influenced by description as well as visual aspects, when choosing an accommodation. Airbnb comes with listings' detailed description and picture. To extract additional information from this form of data might also be prosperous for our model performance. Speaking about dataset size increment and variety of data, further work should also consider a research in the area or

systems and methods, used for processing of large amount of structured as well unstructured data.

Looking at our problem with different perspective, rather than trying to estimate exact price, our research could consider listing's value as a classification problem, where price will be divided into segments. Not to forget about the Airbnb's dynamic pricing, another way of examining problem might be as a time series task. There are many possible ideas for further work, which have a strong application potential. We believe that our research brought better understanding of the field of study as well as its further potentials.

6 Bibliography

[1] S. Ross, „Investopedia," June 2015. [Online]. Available: https://www.investopedia.com/ask/answers/063015/how-much-global-economy-comprised-real-estate-sector.asp.

[2] N. Schriever, „http://bluewatercredit.com/," March 2018. [Online]. Available: http://bluewatercredit.com/ranking-biggest-industries-us-economy-surprise-1/.

[3] S. L. Lynch, „Bloomberg," October 2008. [Online]. Available: https://www.bloomberg.com/news/articles/2008-10-02/metro-u-s-home-prices-fall-on-higher-foreclosures.

[4] Trulia, „https://www.trulia.com/blog/trends/own-to-rent/," August 2016. [Online]. Available: https://www.trulia.com/blog/trends/own-to-rent/.

[5] J. Hamari, S. Mimmi und A. Ukkonen , „The sharing economy: Why people participate in collaborative consumption," June 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23552.

[6] Airbnb, „Airbnb Newsroom," March 2018. [Online]. Available: https://press.atairbnb.com/fast-facts/.

[7] E. Huet , „Forbes," June 2015. [Online]. Available: https://www.forbes.com/sites/ellenhuet/2015/06/05/how-airbnb-uses-big-data-and-machine-learning-to-guide-hosts-to-the-perfect-price/#65391e706d49.

[8] H. Yee und B. Ifrach, „Airbnb Engineering & Data Science," June 2015. [Online]. Available: https://medium.com/airbnb-engineering/aerosolve-machine-learning-for-humans-55efcf602665.

[9] A. Ng, „Machine Learning for a London Housing Price Prediction Mobile Application," Imperial College London, Department of Computing, London, 2015.

[10] D. Epple, L. Quintero und H. Sieg, „Estimating Hedonic Functions for Rents and Values in the Presence of Unobserved Heterogeneity in the Quality for Housing," 2013.

[11] T. Slee, 2017. [Online]. Available: http://tomslee.net/airbnb-data-collection-get-the-data.

[12] M. Cox, „Inside Airbnb,“ 2017. [Online]. Available: http://insideairbnb.com/about.html.

[13] Zillow, „Zillow Research,“ March 2018. [Online]. Available: https://www.zillow.com/research/about-us/.

[14] Zillow Rentals Team, „Rentals Resource Center,“ March 2018. [Online]. Available: http://www.easybib.com/cite/eval?url=https%3A%2F%2Fwww.zillow.com%2Frental-manager%2Fresources%2Fwhat-is-the-rent-zestimate%2F.

[15] Zillow Research, 2018. [Online]. Available: https://www.zillow.com/research/zestimate-forecast-methodology/.

[16] J. Foley, December 2017. [Online]. Available: https://www.discoverlosangeles.com/press-releases/facts-about-los-angeles.

[17] Junior Worldmark Encyclopedia of World Cities , December 2017. [Online]. Available: https://www.encyclopedia.com/places/united-states-and-canada/us-political-geography/los-angeles.

[18] CoreLogic, „California Home Sale Activity by City,“ 2018.

[19] apartment list, February 2018. [Online]. Available: https://www.apartmentlist.com/ca/los-angeles.

[20] aibnbcitizen, March 2018. [Online]. Available: https://www.airbnbcitizen.com/airbnbs-economic-impact-in-los-angeles-in-2017/.

[21] Provalis Research, June 2017. [Online]. Available: https://provalisresearch.com/blog/brief-history-machine-learning/.

[22] A. Hern, „The Guardian,“ June 2014. [Online]. Available: https://www.theguardian.com/technology/2014/jun/09/what-is-the-alan-turing-test.

[23] W. Knight, „MIT Technology Review,“ October 2016. [Online]. Available: https://www.technologyreview.com/s/602744/ibms-watson-is-everywhere-but-what-is-it/.

[24] B. Marr, „Forbes,“ August 2017. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2017/08/08/the-amazing-ways-how-google-uses-deep-learning-ai/#7caee5df3204.

[25] Facebook Research, March 2018. [Online]. Available: https://research.fb.com/category/machine-learning/.

[26] S.-S. Shai und B.-D. Shai, Understanding Machine Learning: From Theory to Algorithms, New York: Cambridge University Press, 2014.

[27] D. Soni, „Supervised vs. Unsupervised Learning,“ April 2018. [Online]. Available: https://www.kdnuggets.com/2018/04/supervised-vs-unsupervised-learning.html.

[28] T. Hastie und R. Tibshirani, June 2016. [Online]. Available: https://lagunita.stanford.edu/courses/HumanitiesSciences/StatLearning/Winter2016/about.

[29] N. Castle, February 2018. [Online]. Available: https://www.datascience.com/blog/what-is-semi-supervised-learning.

[30] R. S. Sutton und A. G. Barto, Reinforcement Learning: An Introduction, 2017.

[31] MathWorks, 2018. [Online]. Available: https://www.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html.

[32] V. Kurama, January 2018. [Online]. Available: https://towardsdatascience.com/supervised-learning-with-python-cf2c1ae543c1.

[33] M. N. S. S. Ke-Lin Du, Neural networks and statistical learning, London: Springer, 2016.

[34] S. Shukla, 2018. [Online]. Available: https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/.

[35] M. T. Leung, H. Daouk und C. An-Sing, „Forecasting stock indices: a comparison of classification and level estimation models',“ 2000.

[36] K. Moore und C. Williams, „Brilliant.org,“ February 2018. [Online]. Available: https://brilliant.org/wiki/classification/.

[37] J. Brownlee, „Machine Learning Mastery,“ December 2017. [Online]. Available: https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/.

[38] V. Powell und L. Lehe, „Explained Visually,“ May 2015. [Online]. Available: http://setosa.io/ev/ordinary-least-squares-regression/.

[39] Xlstat, January 2017. [Online]. Available: https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols.

[40] P. Bruce und A. Bruce, Practical Statistics for Data Scientists, O'Reilly Media, 2017.

[41] P. Gupta, November 2017. [Online]. Available: https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a.

[42] STHDA, March 2018. [Online]. Available: http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/153-penalized-regression-essentials-ridge-lasso-elastic-net/.

[43] G. James, D. Witten und T. H. Robe, An Introduction to Statistical Learning, New York: Springer, 2014.

[44] A. Kassambara, Machine Learning Essentials: Practical Guide in R, STHDA, 2017.

[45] J. Shubham, June 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/.

[46] D. Kane, „Lecture: Data Science - Part XII - Ridge Regression, LASSO, and Elastic Nets,“ 2015.

[47] K. Nishida, March 2017. [Online]. Available: https://blog.exploratory.io/introduction-to-extreme-gradient-boosting-in-exploratory-7bbec554ac7.

[48] L. Pasanen, L. Holmstrom und M. J. Sillanpaa, „Supporting Information for 'Bayesian LASSO, scale space and decision making in association genetics," Oulu, 2015.

[49] scikit-learn, 2018. [Online]. Available: http://scikit-learn.org/stable/modules/linear_model.html#bayesian-ridge-regression.

[50] S. Sayad, „Support Vector Machine," 2018. [Online]. Available: http://www.saedsayad.com/support_vector_machine.htm.

[51] MathWorks, „Understanding Support Vector Machine Regression," 2017. [Online]. Available: https://www.mathworks.com/help/stats/understanding-support-vector-machine-regression.html.

[52] N. Tomuro, „Brief Introduction to Support Vector Machines," DePaul University, Chicago, 2017.

[53] N. Ismail, „Information Age," June 2017. [Online]. Available: http://www.information-age.com/machine-learning-demystified-importance-data-123466738/.

[54] S. Raschka, June 2016. [Online]. Available: https://sebastianraschka.com/blog/2016/model-evaluation-selection-part1.html.

[55] S. Deviant, 2017. [Online]. Available: http://www.statisticshowto.com/probability-and-statistics/descriptive-statistics/.

[56] J. J. Filliben und A. Heckert, „Exploratory Data Analysis," in *Handbook of Statistical Methods*, NIST, SEMATECH, 2013.

[57] J. Brownlee , December 2013. [Online]. Available: https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/.

[58] S. Kotsiantis, D. Kanellopoulos und P. Pintelas, „Data Preprocessing for Supervised Learning,“ Bd. 1, Nr. N. 2, 2006.

[59] C. Bishop, Pattern recognition and machine learning, Berlin: Springer, 2006.

[60] T. Stephens, January 2014. [Online]. Available: http://trevorstephens.com/kaggle-titanic-tutorial/r-part-4-feature-engineering/.

[61] EliteDataScience, February 2018. [Online]. Available: https://elitedatascience.com/feature-engineering-best-practices.

[62] S. Asaithambi, December 2017. [Online]. Available: https://medium.com/greyatom/why-how-and-when-to-scale-your-features-4b30ab09db5e.

[63] B. A. Dũng , March 2016. [Online]. Available: http://dungba.org/why-should-we-implement-feature-scaling-mostly-all-the-time/.

[64] scikit-learn, „Importance of Feature Scaling - scikit-learn 0.19.1 documentation,“ 2017. [Online]. Available: http://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html.

[65] S. Raschka, July 2014. [Online]. Available: http://sebastianraschka.com/Articles/2014_about_feature_scaling.html.

[66] S. Asaithambi, January 2018. [Online]. Available: https://towardsdatascience.com/why-how-and-when-to-apply-feature-selection-e9c69adfabf2.

[67] R. Kern, „Lecture: Feature Engineering Knowledge Discovery and Data Mining 1,“ ISDS, TU Graz, Graz, 2017.

[68] S. Kaushik, „Introduction to Feature Selection methods with an example (or how to select the right variables?),“ December 2016. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/.

[69] R. Kohavi und G. H. John , „Wrappers for feature subset selection," Mountain View, 1997.

[70] Y. Kuang, „A Comparative Study on Feature Selection Methods and Their Applications in Causal Inference," Department of Computer Science, Faculty of Science, Lund University, 2009.

[71] D. F. Gillies und Z. M. Hira, „A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data," Department of Computing, Imperial College London, London, 2015.

[72] V. Powell und L. Lehe, 2015. [Online]. Available: http://setosa.io/ev/principal-component-analysis/.

[73] M. Richardson, „Principal Component Analysis," 2009.

[74] A. M. Martinez und A. C. Kak, „PCA versus LDA," Bd. 3, 2001.

[75] M. Magnani, „Techniques for Dealing with Missing Data in Knowledge Discovery Tasks," 2004.

[76] K. Grace-Martin, 2018. [Online]. Available: https://www.theanalysisfactor.com/causes-of-missing-data/.

[77] A. Swalin, January 2018. [Online]. Available: https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4.

[78] C. Zaiontz, „Real Statistics Using Excel," January 2016. [Online]. Available: http://www.real-statistics.com/handling-missing-data/types-of-missing-data/.

[79] M. Bland, An Introduction to Medical Statistics, Oxford University Press, 2015.

[80] Statistics Solutions, „Statistics Solutions," April 2016. [Online]. Available: http://www.statisticssolutions.com/missing-data-listwise-vs-pairwise/.

[81] K. Maladkar, „Analytics India Magazine," February 2018. [Online]. Available: https://analyticsindiamag.com/5-ways-handle-missing-values-machine-learning-datasets/.

[82] Y. Obadia, January 2017. [Online]. Available: https://towardsdatascience.com/the-use-of-knn-for-missing-values-cf33d935c637.

[83] Y. C. Yuan, „Multiple Imputation for Missing Data: Concepts and New Development," SAS Institute Inc., Rockville, 2016.

[84] S. Santoyo, September 2017. [Online]. Available: https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561.

[85] C. Gorrie, March 2016. [Online]. Available: http://colingorrie.github.io/outlier-detection.html.

[86] J. Brownlee, „Machine Learning Mastery," March 2018. [Online]. Available: https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data/.

[87] F. Tseng, 2016. [Online]. Available: https://frnsys.com/ai_notes/machine_learning/model_selection.html.

[88] Udacity, „Machine Learning Engineer Nanodegree Program," 2017. [Online]. Available: https://www.udacity.com.

[89] R. West, October 2016. [Online]. Available: https://content.nexosis.com/blog/training-set-vs.-test-set.

[90] D. Nautiyal, November 2017. [Online]. Available: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/.

[91] EliteDataScience, February 2018. [Online]. Available: https://elitedatascience.com/overfitting-in-machine-learning#overfitting-vs-underfitting.

[92] P. Refaeilzadeh, L. Tang und H. Liu, „Cross-Validation," Arizona State University, 2008.

[93] A. Bronshtein, May 2017. [Online]. Available: https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6.

[94] M. Kuhn, Applied Predictive Modeling, New York: Springer , 2016.

[95] scikit-learn, 2017. [Online]. Available: http://scikit-learn.org/stable/modules/grid_search.html.

[96] S. Deviant, January 2017. [Online]. Available: http://www.statisticshowto.com/rmse/.

[97] E. Rieuf, January 2017. [Online]. Available: https://www.datasciencecentral.com/profiles/blogs/regression-analysis-how-do-i-interpret-r-squared-and-assess-the.

[98] B. K. Stone, May 2013. [Online]. Available: http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit.

[99] R. Ng, May 2017. [Online]. Available: http://www.ritchieng.com/machine-learning-evaluate-linear-regression-model/.

[100] Tableau Software, „Tableau Software," March 2018. [Online]. Available: https://www.tableau.com/about.

[101] S. Canova, L. Diego und F. Cortinovis, AME Publishing Company, June 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5506131/.

[102] V. Wong, June 2017. [Online]. Available: https://www.buzzfeed.com/venessawong/airbnb-mega-hosts?utm_term=.wakQO6QRZp#.esDvR0vDV1.

[103] G. Seif, February 2018. [Online]. Available: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68.

[104] Los Angeles Times, 2018. [Online]. Available: http://maps.latimes.com/neighborhoods/neighborhood/list/.

[105] J. Vasquez, „SymplyAnalytics," June 2017. [Online]. Available: https://simplyanalytics.zendesk.com/hc/en-us/articles/204848916-What-is-a-census-tract-.

[106] V. Fedak, February 2018. [Online]. Available: https://towardsdatascience.com/big-data-what-is-web-scraping-and-how-to-use-it-74e7e8b58fd6.

[107] DATA.gov, March 2018. [Online]. Available: https://data.lacity.org/A-Prosperous-City/Active-restaurant-heat-map/pwji-zbmi.

List of Figures

List of Tables

List of Abbreviations

KNN	K Nearest Neighbors
OLS	Ordinary Least Square
XGBoosting	Extreme Gradient Boosting
CV	Cross Validation
EDA	Exploratory Data Analysis
PCA	Principal Component Analysis
LOOCV	Leave-One-Out Cross-Validation
RMSE	Root Mean Squared Error
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
MSE	Mean Squared Error
MAE	Mean Absolute Error
SVM	Support Vector Machine
SVR	Support Vector Regressor