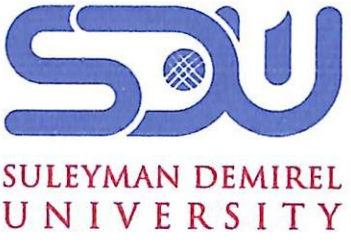
Ministry of Science and Higher Education of the Republic of Kazakhstan

Suleyman Demirel University



Aidana Nametbayeva

Automating real estate investment analysis in Kazakhstan

THESIS

Presented in Partial Fulfilment for the

Master of Technical Sciences Degree in Computer Science

(degree code: 7M06102)

Department of Computer Science

Faculty of Engineering and Natural Sciences

Supervisor: **Kamila Orynbekova**

Kaskelen 2023

Suleyman Demirel University
Faculty of Engineering and Natural Sciences
Department of Computer Science

Dean of Faculty

Associate Professor

PhD Zhamanov A.

06 06 2023

Topic of the thesis:

Automating real estate investment analysis in Kazakhstan

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science" SDU, 2021-2023

Head of Department _____ Assistant Professor, PhD Mukash Zh.

Academic Supervisor _____ Kamila Orynbekova

Master student _____ Aidana Nametabyeva

Kaskelen 2023

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Aidana Nametbayeva

2023

Acknowledgements

I would like to extend my profound thanks to my advisor, Kamila Orynbekova, whose invaluable insights, attention, and patience were indispensable. Her in-depth knowledge of the subject matter and unyielding support laid the groundwork for this research.

I wish to convey my heartfelt appreciation to my family and friends, whose unwavering support, patience, and belief in me has been unfaltering. The love and encouragement I received from you have served as my motivation and inspiration.

Lastly, I wish to acknowledge my sincere gratitude to all the individuals who participated in this study, as without their involvement, this research wouldn't have come to fruition.

Dedication

This research is devoted to my parents, Sabitzhan Nametbayev, Kulzipa Nametbayeva, Nurlan Sadyrbaev, Bibigul Yeshenkulova, who have continuously believed in me and endorsed my quest for knowledge. Their infinite affection, support, and patience have been my source of inspiration and fortitude in the most challenging times.

I would also like to dedicate this study to my partner, Akimali Yeskhozhaev, who has been my rock throughout this journey, offering invaluable support and encouragement, always having faith in my capabilities.

Lastly, this research is dedicated to all the students and researchers with a burning desire to delve into the realms of science and technology. I hope that this study serves as a catalyst to inspire you and aid in your individual explorations.

Abstract

In the modern world, data plays the most important role not only for the development of medium and small businesses and the formation of large economies, but also for the comfortable everyday life of every person. This study examines the Kazakh residential real estate market in the city of Almaty to apply machine learning algorithms to real estate data for price prediction. We know that this market is an important sector of the global economy. The residential real estate market in Kazakhstan is a complex structure, consisting of hundreds of thousands of apartments, characterized by many features. At the same time, any changes in the market may cause speculation and a deliberate increase in real estate prices. That is why it is so important to understand what the real cost of an apartment is and where it is more expensive. The research focused on building a real estate value prediction model using machine learning methods such as linear regression, XGBoost, Random Forest, and Prophet. The data was taken from the krisha.kz website for the following dates: November 30, 2020, May 9, 2021 and November 2, 2022. The number of processed data for each of these dates was 6807, 3426 and 32424 records, respectively. Among the tested models, XGBoost and Random Forest showed the best results.

Аңдатпа

Қазіргі әлемде деректер орта және шағын бизнесті дамыту және ірі экономиканы қалыптастыру үшін ғана емес, сонымен қатар әрбір адамның жайлы күнделікті өмірі үшін ең маңызды рөл атқарады. Бұл зерттеу бағаны болжау үшін жылжымайтын мүлік деректеріне машиналық оқыту алгоритмдерін қолдану үшін Алматы қаласындағы қазақстандық жылжымайтын мүлік нарығын зерттейді. Бұл нарық әлемдік экономиканың маңызды секторы екенін білеміз. Қазақстандағы тұрғын үй жылжымайтын мүлік нарығы көптеген ерекшеліктерімен сипатталатын жүздеген мың пәтерлерден тұратын күрделі құрылым болып табылады. Бұл ретте нарықтағы кез келген өзгерістер алып-сатарлық пен жылжымайтын мүлік бағасының әдейі көтерілуіне себеп болуы мүмкін. Сондықтан пәтердің нақты құны қандай және қай жерде қымбатырақ екенін түсіну өте маңызды. Зерттеу сызықтық регрессия, XGBoost, Random Forest және Prophet сияқты машиналық оқыту әдістерін қолдана отырып, жылжымайтын мүлік құнын болжау моделін құруға бағытталған. Деректер krisha.kz сайтынан келесі күндерге алынды: 2020 жылдың 30 қарашасы, 2021 жылдың 9 мамыры және 2022 жылдың 2 қарашасы. Осы күндердің әрқайсысы үшін өңделген деректер саны тиісінше 6807, 3426 және 32424 жазбаны құрады. Тексерілген үлгілердің ішінде XGBoost және Random Forest ең жақсы нәтиже көрсетті.

Аннотация

В современном мире данные играют наиболее важную роль не только для развития среднего и малого бизнеса и формирования крупных экономик, но и для комфортной повседневной жизни каждого человека. В данном исследовании рассматривается рынок жилой недвижимости Казахстана в городе Алматы для применения алгоритмов машинного обучения к данным о недвижимости для прогнозирования цен. Мы знаем, что этот рынок представляет собой важный сектор глобальной экономики. Рынок жилой недвижимости Казахстана - это сложная структура, состоящая из сотен тысяч квартир, характеризующихся многими особенностями. В то же время любые изменения на рынке могут вызвать спекуляции и намеренное повышение цен на недвижимость. Вот почему так важно понимать, какова реальная стоимость квартиры и где она дороже. Исследование было направлено на создание модели прогнозирования стоимости недвижимости с использованием методов машинного обучения, таких как линейная регрессия, XGBoost, случайный лес (Random Forest), и Prophet. Данные были взяты с сайта krisha.kz на следующие даты: 30 ноября 2020 года, 9 мая 2021 года и 2 ноября 2022 года. Количество обработанных данных для каждой из этих дат составляло соответственно 6807, 3426 и 32424 записей. Среди протестированных моделей XGBoost и Random Forest показали наилучшие результаты.

Abbreviations

- ML - Machine Learning
- AI - Artificial Intelligence
- XGB - XGBoost
- LR - Linear Regression
- RF - Random Forest
- RMSE - Root Mean Square Error
- MSE - Mean Squared Error
- MAE - Mean Absolute Error
- OHE - One Hot Encoding
- DF - Data Frame
- NN - Neural Network
- DL - Deep Learning
- ANN - Artificial Neural Network
- CNN - Convolutional Neural Network
- RNN - Recurrent Neural Network
- LSTM - Long Short-Term Memory
- API - Application Programming Interface
- CSV - Comma-Separated Values

- SQL - Structured Query Language
- IoT - Internet of Things
- GBM - Gradient Boosting Machine
- NLP - Natural Language Processing
- PCA - Principal Component Analysis
- ROC - Receiver Operating Characteristic
- AUC - Area Under Curve

Table of Contents

Chapter 1

Background and motivations

1.1 Introduction

In the current dynamics of the global economy, real estate continues to play a key role, providing stability and confidence in the future. Real estate is one of the main assets for investment and prudent management of this asset can lead to significant returns. In the context of Kazakhstan, and in particular its largest city of Almaty, the specifics of the real estate market represents a unique field for the application of advanced technologies and algorithms to optimize investment decisions. This master's thesis is devoted to the development and application of an algorithm for determining the price of real estate intended for investment in the city of Almaty, Kazakhstan. This study seeks to leverage the principles of machine learning to process data from the real estate sector, with the ultimate aim of pinpointing the most likely property values. Such information can then inform investment strategies.

We posit that the integration of machine learning techniques with a detailed examination and classification of property data can greatly improve the accuracy of property price forecasting. This in turn could simplify the process of making investment decisions. A key goal of this research is to develop a practical tool for investors and property professionals. This tool would ease the process of appraising and choosing properties for investment in the Kazakh market.

It's worth emphasizing that despite the existence of a large volume of research focusing on forecasting real estate prices, very few dive into the specific challenges of the Kazakhstan market. As such, this research is designed to augment understanding and analyses of the trends and patterns in Almaty's real estate market, while setting a foundation for further scholarly pursuits in this area. The object of research of this work is the residential real estate market of the Kazakhstan in the city of Almaty The subject of research is the use of machine learning methods to predict the value of residential real estate. The aim of this work is to create a model for predicting the value of residential real estate, created using machine learning methods. The practical significance of the research results is that as a result of the work, we will receive an up-to-date value prediction model for residential real estate objects, which can be used to determine the value of residential property of the Kazakhstan.

1.2 Literature review

In the past five years, the amount of research on the application of machine learning in the field of real estate has increased significantly, indicating the rapid development of this field. The article "A Data Mining-Based Real Estate Investment Analysis Framework." by Huang et al. (2019) proposes a real estate investment analysis framework that uses data mining techniques to identify profitable real estate investment opportunities and improve investment decisions[1]. The framework consists of four main steps: data preparation, feature selection, model construction, and investment decision-making. In the data preparation step, the authors collected data on various factors that affect real estate investments, such as economic indicators, demographic factors, and real estate market conditions. The data was then cleaned and preprocessed for further analysis. In the feature selection step, the authors used principal component analysis (PCA) to select the most relevant features for predicting real estate investment returns. The authors found that the most important factors were GDP growth rate, interest rate, housing price index, and population density. In the model construction step, the authors used machine learning algorithms, such as random forest and support vector regression, to predict real estate investment returns based on the

selected features. The authors compared the performance of the machine learning models with traditional methods, such as the discounted cash flow (DCF) model and the net present value (NPV) model. The results showed that the machine learning models outperformed the traditional methods in predicting real estate investment returns. The authors also conducted a sensitivity analysis to determine the impact of different factors on investment returns. In the investment decision-making step, the authors used the results from the model construction step to make investment decisions. The authors suggested that the framework could help identify profitable real estate investment opportunities and improve investment decisions.

Pagoras (2019) and colleagues conducted a think about that utilized machine learning calculations to anticipate genuine domain costs in Chicago, USA. Amid the think about, they connected a few models, counting relapse, irregular timberland, and angle boosting, and concluded that choice tree-based models given the finest exactness[2]. A consider by Jean and Li (2020) inspected different angles of the application of machine learning to genuine bequest, counting its application to cost forecast and advertise slant examination. They emphasized that machine learning can offer assistance overcome the issue of the impact of different components on the esteem of genuine domain, since models can take into consideration a huge number of factors[3]. Petrov and Makeev (2021) investigated the conceivable outcomes of applying machine learning calculations to genuine domain valuation within the Russian advertise. The think about appeared that machine learning can give profoundly precise property cost forecasts, indeed in complex and heterogeneous markets such as the Russian market[4]. Nguyen et al. (2018) examined the application of machine learning to property cost determining in Singapore. They found that convolutional neural systems and irregular woodland perform way better than other machine learning models. Tien and Wu (2019) created a machine learning show for genuine domain esteem forecast within the Taiwan showcase. Their investigate illustrates how machine learning can make strides the precision of forecasts, empowering more precise speculation choices[5]. Brett and Marshall (2019) investigated the application of profound learning to genuine domain cost expectation in Australia. Their comes about appear that

profound learning can give noteworthy enhancements in expectations compared to conventional strategies[6]. Zeng and Jiang (2020) analyzed the application of different machine learning strategies to genuine domain valuation in China. They concluded that it is critical to utilize the proper machine learning demonstrate for each particular case[7].

Kim and Kim (2020) did research on the use of machine learning for South Korean real estate price forecasting. They discovered that employing gradient boosting might produce extremely precise predictions[8].

Strous and Moore (2021) investigated how machine learning could be used to value real estate in the US. Their findings demonstrated that machine learning algorithms can produce predictions that are more precise and effective[9].

Burton and Garcia (2021) investigated the application of machine learning to forecast Spanish real estate prices. Modern machine learning models can considerably increase the accuracy of real estate price projections, according to their findings[10].

In the UK, Raun (2022) used machine learning to analyze real estate data. They discovered that using machine learning algorithms can aid in identifying crucial elements that influence real estate values[11].

Neural networks were employed by Fernandez and Gomez (2022) to forecast Brazilian real estate prices. Their research emphasized the significance of picking the appropriate neural network architecture to guarantee the highest degree of prediction accuracy [12].

In Germany, Becker and Schmidt (2023) investigated the use of machine learning algorithms to forecast real estate prices. When compared to conventional methods, they discovered that machine learning techniques can significantly increase forecast accuracy[13].

Finally, a study by Kim and Park (2022) reveals how deep learning may be used to predict real estate prices. They processed and analyzed massive amounts of real estate data using neural networks, and the results demonstrated that deep learning techniques can significantly outperform more conventional techniques in

terms of forecast accuracy [14].

Chapter 2

Data Collection

2.1 Data Collection

In an era of continuously evolving globalization processes, when big cities become the main drivers of economic growth and innovation, understanding the dynamics of their development is becoming increasingly important. One of the key indicators of urban development is real estate - a reflection of economic activity, the level of prosperity and attractiveness of the region. In my dissertation, I will focus on real estate price forecasting in Almaty, a city that in many ways is a symbol of urban growth and progress in Kazakhstan.

Almaty, being the largest city of Kazakhstan and its former capital, plays an important role in the country's economy. The largest financial institutions and many trading companies are concentrated in Almaty, which makes the city an important economic center. Due to its geographical location and cultural heritage, Almaty is also an important tourist and cultural center.

I begin my work with an analysis of the dynamics of real estate prices in Almaty. The study of real estate prices will allow us to better understand urban development, as well as predict economic trends. This work has practical implications for investors, the government and people who want to move to Almaty.

Real estate price data from the last few years will be used and several machine learning methods will be applied to build price prediction models. I hope that

my findings and suggestions will help city planners, real estate developers and investors in their decisions. The purpose of this study was to create a model for predicting real estate prices based on data from the Kazakhstani real estate website krisha.kz [15]. Data was collected at three different time points: November 30, 2020, May 9, 2021, and November 2, 2022, to ensure data diversity and objectivity for subsequent analysis.

The method of parsing data from the site krisha.kz was used [15]. Parsing was carried out for each district of the city of Almaty separately, as well as taking into account the number of rooms in each residential building. This approach allowed us to obtain more detailed information about the cost of housing in different areas and with different numbers of rooms [16].

After the parsing step, all collected data was combined into one CSV file. This was done to simplify the process of subsequent analysis and data processing. The CSV file provides a versatile and easily accessible format for storing and processing large amounts of data.

As a result of this process, an extensive data set was created, which includes detailed information on the value of real estate in different areas of the city of Almaty over different periods of time. This data will be used to train and test machine learning models, which will then be used to predict property prices. For parsing data from the chosen website we used three different libraries: requests, BeautifulSoup and csv. The first function get-html refers to the website page in order to obtain the content of the page (html).We used get functions from requests library and it returned a response of 200. Well, that means that we have already accessed the site. The second significant function is get-content. Using this function gives us an opportunity to take the content from the specified html. Here we use BeautifulSoup for parsing html and get access to the page elements. Also we used find-all function from BeautifulSoup, which searches for all elements of the class='a-card-inc'. In this function get-content, we create a loop in order to write the elements to the cards[] array. To make this happen, we just take the append and find functions. The function get-content returns the array (cards[]). The third function we want to tell you about is save-doc function. It saves everything we paired in a file. And here we use the csv library and

the functions writerow, writer.Using the loop, we write down all the elements. The most recent and important function is parser. We use the PAGINATION constant here. And we would like to present a part of our code below in figure 1. parser function: Start Ask the user to enter the number of pages (PAGENATION) Convert PAGENATION to integer Get HTML using URL If the HTML response status is 200: Create an empty list called "cards" For each page in the range from 1 to PAGENATION: Display current page number Get HTML using URL and 'page' parameter equal to current page number Append HTML-derived content to "cards" list Save "cards" to CSV file Output "cards" Otherwise: Print 'Error' End

The second step is Merge data from each district into one CSV. For merging data we used 2 libraries:os,pandas. Firstly, read a single csv file by readcsv function. Take list all files in a directory by listdir function. Concatenating multiple csvs together to create a new DataFrame (concat). Reading in Updated dataframe.

Chapter 3

Data Preparation 3

3.1 Data preparation

The third step is Clean up the data! Take data (see Figure 3.1) from the column 'Main' and add new columns: 'name','area','floor','allfloor'.For this we create function analysisofname,where we use split,append,replace functions.Deleted unnec-

	Заголовок	ссылка	Цена	Адрес	Дата	comment
0	4-комнатная квартира, 100 м²Все заметкиУдалить	https://krisha.kz//a/show/51023039	от 30 000 000 ₸	Туркситский р-н, 100-я улица 542	Алматы9 мая	жил. комплекс Жана Куат (КП), 2 этажа, 2021 г....
1	4-комнатная квартира, 81.7 м², 3/5 этажВсе зам...	https://krisha.kz//a/show/666679942	30 000 000 ₸	Туркситский р-н, Гете 305	Алматы3 мая	панельный дом, 1990 г.п., состояние: хорошее, ...
2	4-комнатная квартира, 101 м², 3/5 этажВсе заме...	https://krisha.kz//a/show/665679861	35 000 000 ₸	Туркситский р-н, улица Тынышбаева — Рихарда Зорге	Алматы6 мая	кирпичный дом, 1993 г.п., состояние: хорошее, ...
3	4-комнатная квартира, 79.3 м², 2/5 этажВсе зам...	https://krisha.kz//a/show/666806138	31 000 000 ₸	Туркситский р-н, мкр Алтай-1, Майлина 109 — Гёте	Алматы8 мая	кирпичный дом, 1990 г.п., состояние: хорошее, ...
4	4-комнатная квартира, 130 м², 16/18 этажВсе за...	https://krisha.kz//a/show/667200398	55 000 000 ₸	Туркситский р-н, мкр Жулдыз-1, Брусиловского	Алматы9 мая	жил. комплекс Алтын Булак 2, монолитный дом, 2...

Figure 3.1: Data

essary data with replace function.And make columns correct type with int. Take data from the column 'Price' and add new columns: 'price'. For this we create a function analysisprice, where we use isnull,append,replacefunctions.Deleted unnecessary data withreplace function. And make the column the correct type with int. Take data from the column 'Address' and add new columns: 'region'. For this we create a function analysisregion,where we use split,append,replacefunctions. Deleted unnecessary data with replace function. And make column correct type. Take data from the column 'comment' and add new columns: 'years'.For this we create a function getyear,where we use split,append functions. And make columns

the correct type. The following pseudo-code is a detailed step-by-step translation of the above code into Python: extract-data(df) function:

1. Creating an empty list 'data'
2. Start looping over each row in df:
3. Retrieving the comment from the current df line and storing it in the 'comment' variable
4. Using a regular expression to find and extract 'state' from a comment, storing the result in 'state'
5. Using a regular expression to find and extract 'ceilings' from a comment, store the result in 'ceiling' with 'm' appended if found, otherwise store 'None'
6. Using a regular expression to search for and extract information about a 'bathroom' from a comment, storing the result in 'bathroom' in the format 'N/N' or 'N' or 'None' if matching groups are found
7. Using a regular expression to search for and extract 'phone' from a comment, storing the result in 'phone'
8. Using a regular expression to search and extract 'internet' from a comment, storing the result in 'internet'
9. Using a regular expression to search for and extract 'kitchen' from a comment, store the result in 'kitchen' with 'm^2' appended if found, otherwise store 'None'
10. Adding a tuple (state, ceiling, bathroom, phone, internet, kitchen) to the 'data' list
11. Convert list 'data' to DataFrame 'df-data' with specific column names
12. Combining the original DataFrame 'df' and the new DataFrame 'df-data', returning the resulting DataFrame.

This pseudocode details the process of extracting specific information from text data using regular expressions, converting the resulting data into a DataFrame

format, and merging the resulting DataFrame with the original data. In this code, this is done for the following categories:

- state: The state of the property.
- ceiling: Ceiling height.
- bathroom: Information about the bathroom.
- phone: Presence of a phone.
- internet: Internet presence.
- kitchen: Kitchen area.

This code is a good example of using regular expressions to extract useful information from text data in a pandas dataframe. This code performs several tasks related to processing and converting dates in a pandas dataframe. Below is a detailed description of each step: Creating the 'month-mapping' dictionary: A dictionary is created that maps Russian month abbreviations to their English equivalents. This is necessary to convert dates presented in Russian into a format that can be easily processed using the pandas library. Function 'convert-date-rus-to-eng(date-str)': This function takes a string with a date in Russian and returns a string with a date in English. To do this, a regular expression is used that extracts a number (day) and a word (month) from the input string, and then uses the 'month-mapping' dictionary for the month to convert the Russian name of the month to English.

Applying the 'convert-date-rus-to-eng(date-str)' function to the 'Date' column: Here the 'convert-date-eng-to-eng(date-str)' function is applied to the entire 'Date' column in the pandas dataframe using the '.apply()' method. This will convert all dates in the 'Date' column from Russian format to English. Convert 'Date' column to datetime format: In this step, the 'Date' column, now containing dates in English, is converted to datetime format using the 'pd.to-datetime()' function. Change year to 2022: Since all dates in the dataframe are in 2022, all datetime objects in the 'Date' column are changed so that the year is 2022. Rename column 'Date' to 'year-sub': Finally, the 'Date' column is renamed to 'year-sub' for clarity. All these operations provide a correct and convenient representation

of dates in the pandas dataframe, which is important for further data analysis. After processing and cleaning the data, our data set is now completely ready for further analysis and use in a machine learning model. The following main steps were performed during data preparation:

1. Processing and transformation of date data from the Russian-language format into a standard English-language format has been carried out, which allows you to work with data using the built-in pandas and Python functions.

2. The specific information contained in the text comments for each property was extracted and converted into a structured format using regular expressions.

3. Consistent replacement of the year with the current one and renaming of the column for greater clarity and clarity of the data was carried out.

4. All of the above activities resulted in a structured and clean dataset ready for further analysis.

Thus, using a combination of various data processing and cleaning techniques, we have successfully converted the original data into a format that can be used for further analysis and modeling. This is a key milestone in any data analysis and machine learning project, and the successful completion of this milestone is the foundation for the success of the next phases of the project.

	area	floor	allfloor	price	region	year	priceth	room	state	ceiling	bathroom	phone	internet	kitchen	year_sub
0	75.0	1	3	35500000	Жетысуский	1984	39.5	3	среднее	2.7 м	None	отдельный	ADSL	None	2022-11-02
1	72.0	1	5	42000000	Жетысуский	1988	42.0	3	хорошее	None	None	есть возможность подключения	None	None	2022-11-02
2	64.0	5	5	35000000	Жетысуский	1973	35.0	3	None	None	None	None	None	None	2022-11-02
3	75.0	3	9	44500000	Жетысуский	2008	44.5	3	хорошее	2.7 м	None	None	ADSL	None	2022-11-02
4	65.4	4	6	31300000	Жетысуский	2015	31.3	3	хорошее	None	None	None	None	None	2022-11-02

Figure 3.2: Prepared data

The data preparation and cleansing process processed three separate data sets, each corresponding to a different time point: November 30, 2020, May 9, 2021, and November 2, 2022. The number of data processed for each of these dates was 6807, 3426 and 32424 records, respectively. The data was extracted from the

real estate website krisha.kz, each area was analyzed separately and depending on the number of rooms, and then all the data was combined into one CSV file for each date. Detailed cleaning and data processing was carried out for each data set in accordance with the procedures described above. The collection of data from different time points allows us to observe the dynamics of real estate prices in the city of Almaty over a certain period of time. A total of 42157 records were processed, which gives us a large amount of data to analyze and predict real estate prices. Now that the data (see Figure 3.2)has been successfully collected, processed, and cleaned, the next step in our research is data visualization. This will allow us to better understand the structure of our data, identify any visible trends or patterns, and help identify key characteristics that may influence real estate prices in the city of Almaty. Visualization is an essential tool in the data analysis process to help us see the big picture and improve our understanding of the data before applying machine learning algorithms.

Chapter 4

Data Visualization 4

4.1 Data Visualization

When we have perfectly structured and parsed data, it is much easier to make some data visualization. Data visualization helps people to understand the data by showing it graphically, using trends and correlations. In this research, we used different libraries for data analysis. Now we need to look at our data, that is, what the prepared data consists of . A pseudocode representing the above Python code and a detailed explanation of each step are given below:

1. Creating a new figure for the graph with the specified dimensions (12,10)
2. Get the number of occurrences of each unique value in the 'region' DataFrame 'df' column using the value-counts() function
3. Building a pie chart based on the obtained values using the plot.pie() function. Using the 'autopct' parameter to specify the display format for the percentage value, which allows the percentage of each value to be displayed on the graph with an accuracy of one tenth of a percent
4. Displaying the generated chart using the show() function

This pseudocode details the process of constructing a pie chart to visualize the distribution of values in a particular DataFrame column. This is useful for visualizing the proportion of each unique value in the total amount of data.

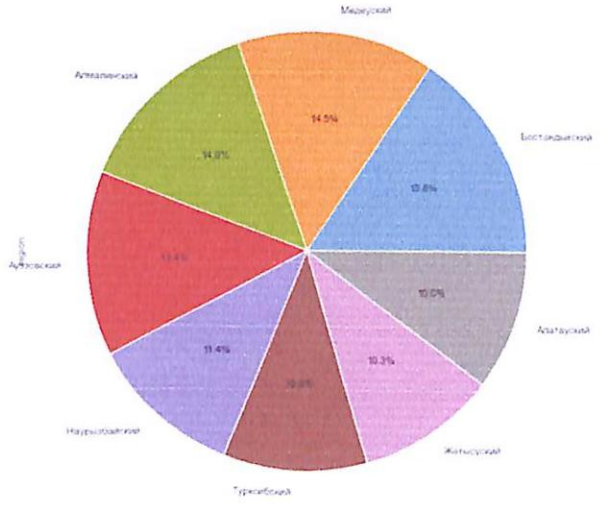


Figure 4.1: Pie chart 2020 year

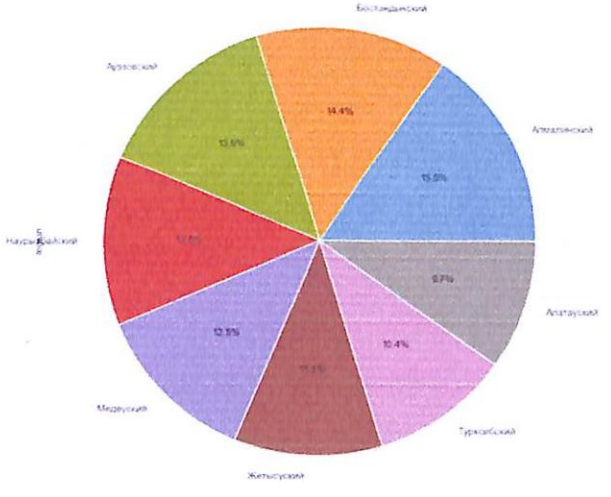


Figure 4.2: Pie chart 2021 year

Analyzing the data obtained, we see interesting trends in the distribution of real estate by districts of the city of Almaty (see Figure 4.1) in the period from 2020 to 2022.

1. Auezov district shows a stable value of about 13-14 %, but in 2022 its share increases to 18.2%. This may indicate an increase in interest in the area or an increase in construction.

2. Bostandyk district has the largest share in 2022, equal to 28.8%, while in 2020 and 2021 its share was about 14-15%. This indicates a significant increase in activity in the real estate market in the area.

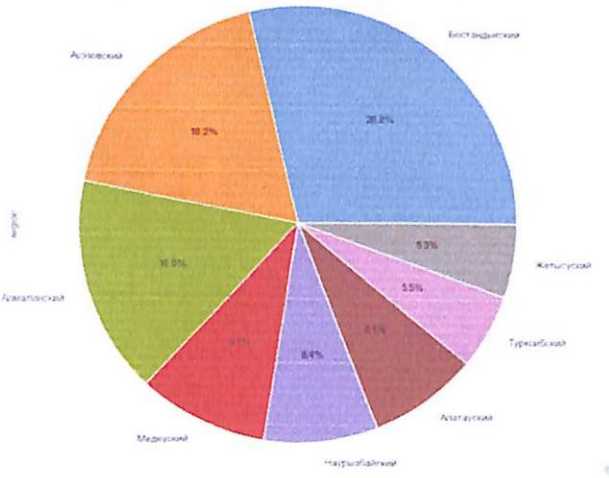


Figure 4.3: Pie chart 2022 year

3. Zhetysu and Turksib regions demonstrate relative stability over all three years with slight fluctuations around 10%.

4. Alatau region also shows stability over all three years with a slight decrease in the share in 2022.

5. Nauryzbai district shows a decrease in share from 12.5% in 2021 (see Figure 4.2)to 8.4% in 2022.

6. Medeu district also shows a decrease in share from 14.5% in 2020 to 9.7% in 2022 (see Figure 4.3).

7. Almaly district , on the contrary, shows an increase in the share from 14.0% in 2020 to 16.0% in 2022.

In general, this data reflects the dynamics and volatility of the Almaty real estate market, and can be useful in understanding which areas are becoming more or less popular among buyers and investors.

The pseudocode and detailed description for this Python code is as follows:

1. Grouping DataFrame 'df' data by 'region' column using the groupby() function. This allows the data to be aggregated according to each unique value in the 'region' column.

2. Calculate the average value of the 'priceth' column for each group obtained

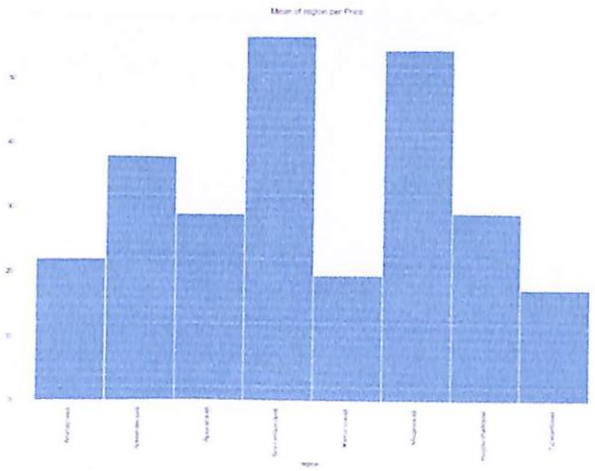


Figure 4.4: Mean of region per price 2020 year

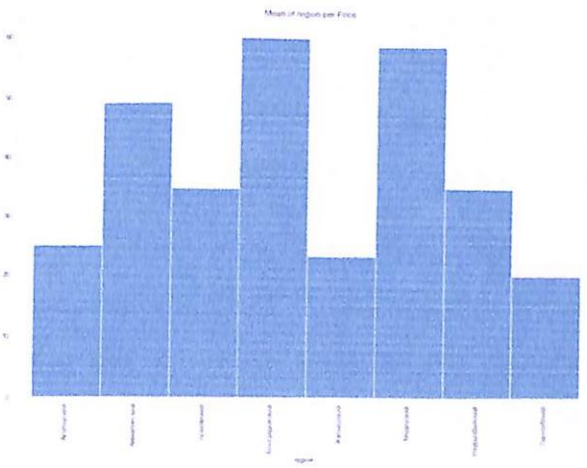


Figure 4.5: Mean of region per price 2021 year

in the previous step using the mean() function.

3. Building a bar chart based on the calculated average values using the plot.bar() function. The column width is set to 1 and the chart size is set to (15,10).

4. Setting the chart title to 'Mean of region per Price' using the title() function.

5. . Displaying the constructed graph using the show() function.

This pseudocode details the process of building a bar chart to visualize the average price values ('priceth' column) for each region ('region' column) in a DataFrame. This is useful for visualizing the dependence of the average price on the region.

1. Medeu district is the most expensive area to purchase real estate during all

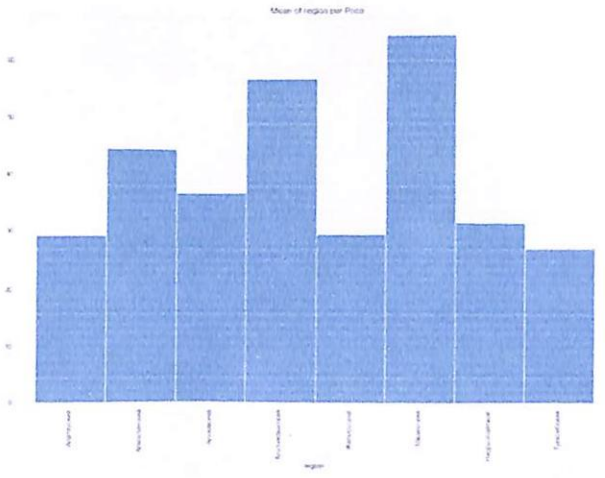


Figure 4.6: Mean of region per price 2022 year

three years under review (2020, 2021 and 2022). In this area, the average cost is 65, 59 and 55 million tenge, respectively. This speaks of the high demand and prestige of the area, which makes it attractive to investors (see Figure 4.4).

2. Bostandyk district ranks second in terms of real estate value, maintaining stable indicators for the period under review (55, 60 and 60 million tenge). The area may also be of interest to investors due to its stability and high prices (see Figure 4.5).

3. At the opposite end of the spectrum, Zhetysu and Turksib districts show the lowest average real estate prices. In these areas, the average cost of real estate ranged from 15 to 25 million tenge (see Figure 4.6).

4. In Auezov, Alatau and Nauryzbai districts, average real estate prices are observed, which fluctuate around 25-35 million tenge. These areas may be of interest to buyers looking for more affordable housing options.

5. Almaly district shows some increase in property value over time, after analyzing the data for 2020, 2021 and 2022 (37, 50 and 45 million tenge, respectively).

In conclusion, this data provides valuable information for investors and buyers who are planning to invest in real estate in the city of Almaty. Through this analysis, they can better understand the dynamics of real estate prices in different

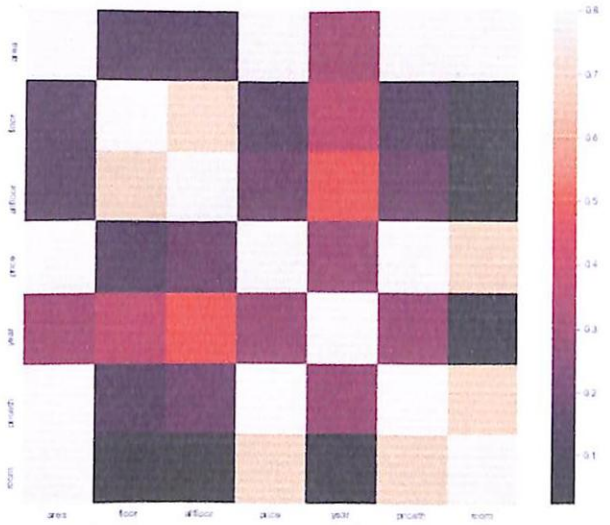


Figure 4.7: Correlation matrix 2020 year

areas of the city. The pseudocode and detailed description for this Python code is as follows:

1. Calculation of the correlation matrix for the DataFrame 'df' using the corr() function. This matrix shows the degree of correlation (connection) between each pair of columns in the DataFrame.

2. Creation of a new graphics window (figure) and axes (axes) using the subplots() function of the matplotlib library. The window size is set to (12, 9).

3. Building a heatmap of the correlation matrix using the heatmap() function of the seaborn library. The maximum value for the color scale is set to 0.8 and the shape of the cells is set to square.

This pseudo-code details the process of creating a heat map for visualizing the correlation matrix. A heat map allows you to visualize the degree of relationship between the various parameters of a dataset. This allows you to identify the most closely related parameters and use this information when building machine learning models or for other analytical purposes. As a result of executing this code, you will get a heat map of correlations between various features in the data. This map will help you quickly see which traits are strongly related to each other.

Based on the results of the correlation analysis, we see a significant relationship

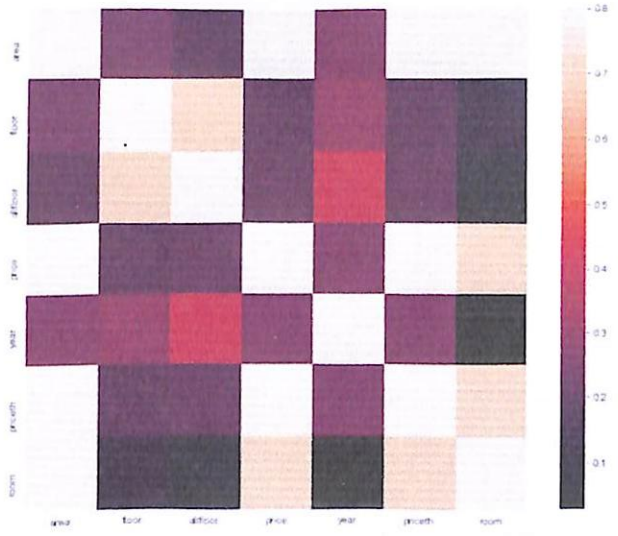


Figure 4.8: Correlation matrix 2021 year

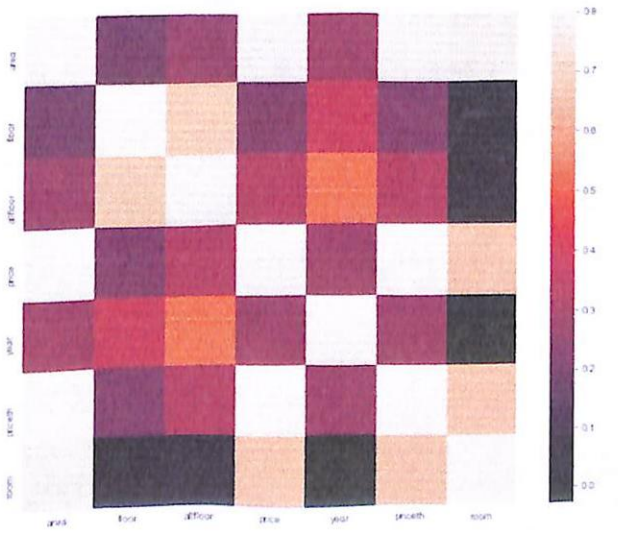


Figure 4.9: Correlation matrix 2022 year

between the price of real estate and parameters such as area and number of rooms. This indicates that with the increase in the size of housing and the number of rooms, the value of real estate in Almaty also increases .

This dependence can be traced equally throughout all the considered periods of time - 2020, 2021 and 2022, which emphasizes its sustainable nature and importance in the formation of the cost of housing in the Almaty market (see Figure 4.7). This is quite logical, since in most cases the area and number of rooms are key factors that are taken into account when determining the price of a property. These results (see Figure 4.8) help us better understand the structure of the city's real estate market and can be useful in developing a real estate price prediction model.

If we take over the dependent variable price for an apartment, that is a good linear dependence with 'area' and 'room'. There is still a direct dependence with 'priceth', but this is understandable because we form this feature as division by a million and it will be necessary to remove it. Another problem is that there is a big correlation between 'area' and 'room' and it can badly affect linear models, so when building a linear regression is to remove, in my opinion, 'room', since it is a more "small" feature and carries far less information 4.9. Here is the pseudocode and detailed description for the above Python code:

1. Create a new 6x6 inch graphic window (figure) using the figure() function of the matplotlib library.

2. Creation of three-dimensional axes using the Axes3D function. The auto-add-to-figure parameter is set to False, which means that the axes are not automatically added to the graphics window.

3. Adding the created axes to the graphics window using the add-axes() method.

4. Write columns 'priceth', 'area', and 'room' from DataFrame 'df' to variables x, y, and z respectively.

5. Creation of a three-dimensional scatterplot using the scatter() method of the axes. Point coordinates are given by x, y, and z values. The color of the dots is determined by the x values. The shape of the marker is set to

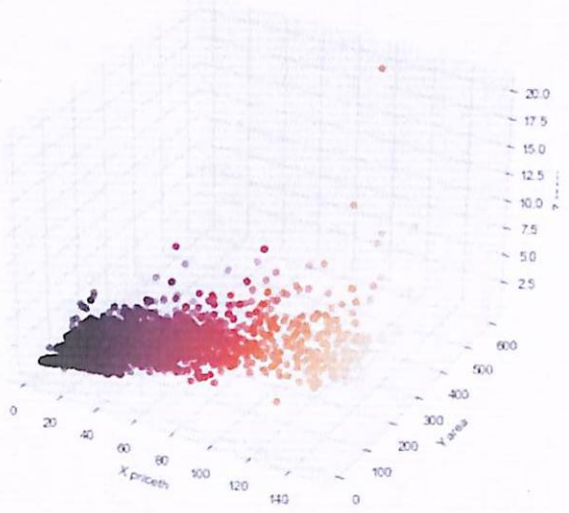


Figure 4.10: 3D scatter plot 2020 year

'o', which corresponds to round dots.

6. Set the x, y, and z axis labels to 'X priceth', 'Y area', and 'Z room' respectively using the set-xlabel(), set-ylabel(), and set-zlabel() methods.

7. Displaying a graph using the show() function of the matplotlib library.

This code creates a 3D scatterplot to visualize relationships between the 'priceth', 'area', and 'room' columns in the DataFrame 'df'. Each point on the graph represents a separate row in 'df', and its x, y, and z coordinates correspond to the 'priceth', 'area', and 'room' values of that row. The point color is determined by the 'priceth' value. As a result, each point on this graph represents a separate property with its price, area and number of rooms. This kind of visualization is very useful for analyzing the relationship between three variables at the same time. Based on our 3D visualizations and analysis, we can conclude that the price of real estate in Almaty has a direct correlation with its area and the number of rooms (see Figure 4.10).

With an increase in the area of an apartment or the number of rooms in it, the price of real estate, as a rule, increases (see Figure 4.11). This is a completely expected result, as large apartments or apartments with more rooms usually cost more due to increased living space and potential living comfort.

The flip side of this pattern is also true: properties with a smaller footprint and

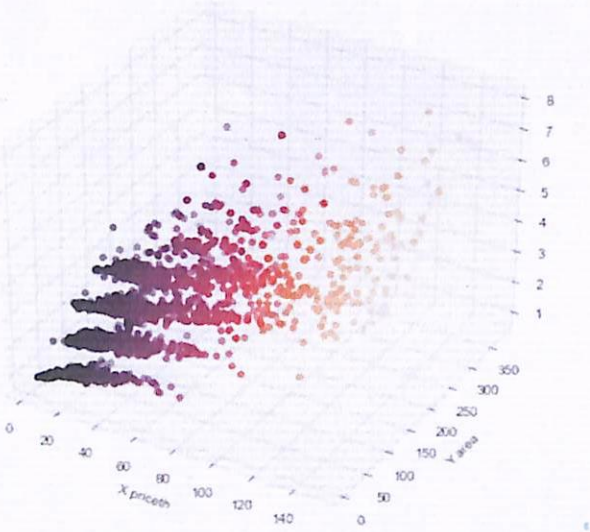


Figure 4.11: 3D scatter plot 2021 year

fewer rooms usually cost less (see Figure 4.12). This reflects a general economic logic: a smaller quantity of a good or service is usually available at a lower price. The findings from our research can be used to determine the expected value of real estate in Almaty based on its area and number of rooms, which in turn can help potential buyers or investors make informed decisions [17].

The pseudocode and detailed description for the above Python code is as follows:

1. Importing the norm function from the scipy.stats module. This function allows you to work with a normal distribution.

2. Importing the stats module from the scipy library. The stats module contains a large number of statistical functions.

3. Creating a histogram for the 'priceth' column from the DataFrame 'df' using the histplot() function of the seaborn library.

4. Creating a new graphics window (figure) using the figure() function of the matplotlib library.

5. Plotting a Q-Q plot (quantile-quantile) for the 'priceth' column from the DataFrame 'df' using the probplot() function from the scipy.stats module. The Q-Q plot allows you to visually assess how the sample data corresponds

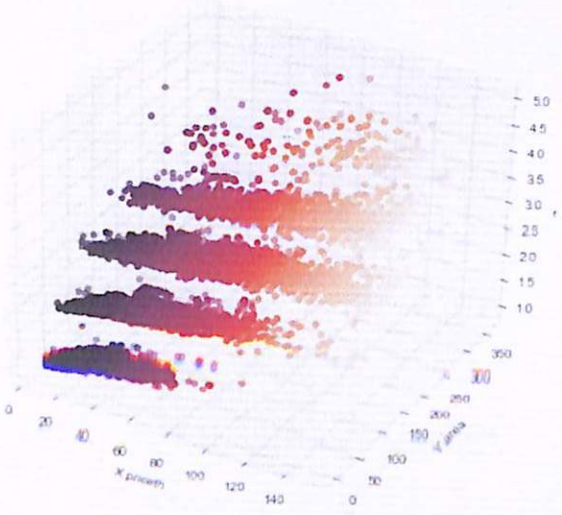


Figure 4.12: 3D scatter plot 2022 year

to the theoretical distribution (in this case, normal).

This code allows you to visualize the distribution of real estate price values ('priceth') using a histogram and a Q-Q plot. The histogram allows you to evaluate the shape, central trend, and spread of the data distribution. The Q-Q plot is intended for visual verification of the conformity of the sample data with the theoretical distribution. If the points on the Q-Q plot are located along a straight line, then this indicates the consistency of the data with the expected distribution. Analyzing real estate price data in the city of Almaty in Kazakhstan, one can notice noticeable changes over the past three years. In 2020 (see Figure 4.13) , residential property prices ranged between 20,000 and 40,000, with 400 to 600 offers on the market. This may indicate a stable market condition during this period, with moderate supply and demand. a In 2021 (see Figure 4.14), we are seeing a slight decrease in the number of offers on the market - from 250 to 400, while prices have remained almost at the same level, fluctuating between 20 and 40 thousand. This may indicate some cooling of the market, while maintaining prices at the same level. However, in 2022 (see Figure 4.15). the situation has changed markedly. Firstly, the number of offers on the market has noticeably increased - from 1,000 to 1,600. Secondly, the price range has also expanded - from 20,000 to 60,000. This may indicate a significant increase in activity in the market, possibly related to general economic growth or changes in the policy of

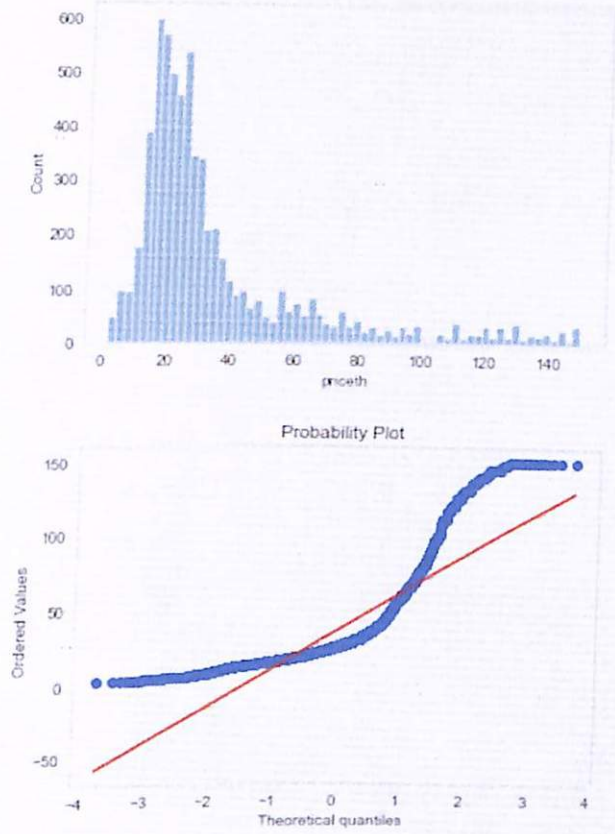


Figure 4.13: Probability plot 2020 year

regulating the real estate market. Overall, these data underline the importance of continuous analysis of the real estate market, as it is constantly changing under the influence of many factors. This is especially important for investors looking to optimize their real estate investment.

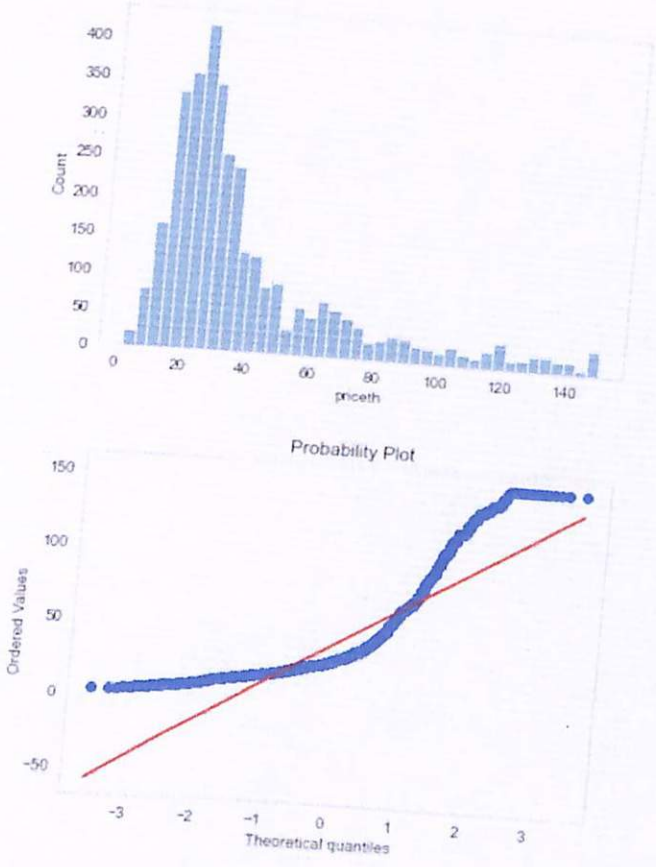


Figure 4.14: Probability plot 2021 year

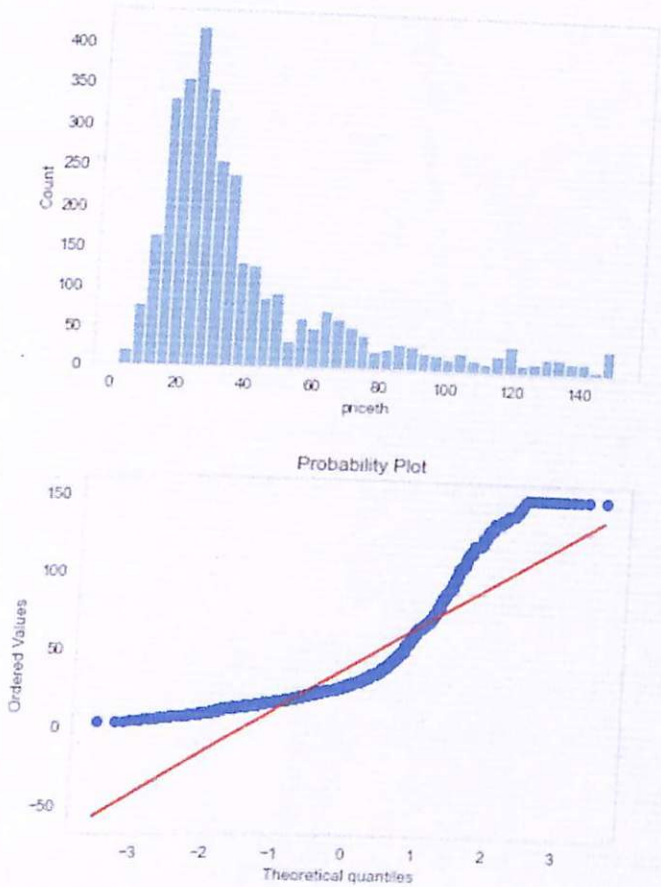


Figure 4.15: Probability plot 2022 year

Chapter 5

Data Analytics 5

5.1 Data Analytics

After carefully looking at and visualizing the collected genuine bequest cost information within the city of Almaty in Kazakhstan, we have come to an imperative point of reference in our investigate. Our work with a dataset that combines data over the past three a long time has permitted us to see patterns, changes and designs that at to begin with look might have gone unnoticed.

In any case, the information alone does not give a total picture. For a more profound understanding and expectation of genuine bequest cost flow, the following step is to construct and prepare a machine learning demonstrate. This will permit us not as it were to more precisely survey the affect of different components on the esteem of genuine domain, but also to foresee long-term improvement of the showcase circumstance.

In this regard, our research enters the next critical stage - data analysis and development of a machine learning model. Our goal is to create a reliable, accurate and efficient model that can serve as a reliable tool for making real estate investment decisions.

We start this exciting journey into the world of machine learning with great enthusiasm and look forward to new discoveries and knowledge.

The pseudocode and detailed description for the above Python code is as follows:

1. Creation of the OneHotEncoder object from the sklearn.preprocessing library. This object will be used to convert categorical variables to binary (0 and 1) values.
2. Transformation of categorical columns from DataFrame 'df' using fit-transform() function.This function applies the one-hot encoding method to the data, training the encoder on the data and then transforming it. The converted data is then converted to a numpy array using the toarray() function.
3. Obtaining feature names after encoding using the get-feature-names-out() function.These names are used to create a new DataFrame.
4. Create a new DataFrame 'encoded-df' using the converted data and feature names.
5. Removing the original categorical columns from DataFrame 'df'.
6. Concatenation of the original DataFrame 'df' and the new DataFrame 'encoded-df' using the pd.concat() function.
7. Printing the first rows of the resulting DataFrame and its columns.

This code is the process of converting categorical variables to numeric using the One-Hot Encoding method. This is an important step in data preparation for many machine learning algorithms since most of them can only work with numerical data. One-Hot Encoding allows categorical data to be represented as binary vectors, making it suitable for use in machine learning algorithms. The pseudocode and detailed description for the above Python code is as follows:

1. Deleting rows with missing values in the 'price' column from the DataFrame 'df' and saving the result in 'df-cleaned'.
2. Defining columns with categorical variables.
3. Creation of the OneHotEncoder object for converting categorical variables into binary (0 and 1) values.

4. Converting categorical columns from 'df-cleaned' using the 'one-hot-encoder' object and storing the results in 'encoded-data'.
5. Creation of DataFrame 'df-encoded' using 'encoded-data' and feature names received from 'one-hot-encoder'.
6. Removing the original categorical columns from 'df-cleaned' and concatenating it with 'df-encoded' using the pd.concat() function.
7. Defining features 'X-cleaned' and target variable 'y-cleaned' by removing the 'price' column from 'df-cleaned'.
8. Create a SimpleImputer object to replace missing values in the data using the average value of each feature.
9. Applying SimpleImputer to 'X-cleaned' and storing the result in 'X-imputed'.
10. Splitting 'X-imputed' and 'y-cleaned' into train and test datasets using the train-test-split() function.
11. Creation and training of a linear regression model on a training data set.
12. Obtaining predictions from the model on a test data set.
13. Calculation of the root mean square error and the coefficient of determination between forecasts and real values.
14. Output of the value of the mean square error and the coefficient of determination.

This code is a complete data preprocessing and training pipeline for a machine learning model. Includes steps such as removing missing values, encoding categorical variables, replacing missing values, separating the data into training and test sets, training the model, and evaluating its performance. In general, this code represents the complete process of training a machine learning model, from data preprocessing to performance evaluation. As part of our analysis, we applied a machine learning model to predict property prices. We used linear regression as our model because it is one of the most common and intuitive models for this kind of problem.

However, the results we got from our model were not satisfactory. Two key metrics used to evaluate the quality of our model showed significant problems.

The mean square error (MSE) of our forecasts was 4.07e+23. This is a very high value, which indicates that our model is producing predictions that, on average, are very different from the real values. In an ideal situation, we would like to see the MSE value go to zero.

The second metric, the coefficient of determination R^2, was -487189872.44273096. This value indicates that our model predicts worse than a model that would simply use the average property price for all predictions. The ideal R^2 value is 1, which means that our model perfectly explains the variation in the data.

These results indicate that we need to rethink our modeling approach. We may need to select a different model, change model parameters, or further process our data to better prepare it for model training. We will continue our work in this direction, striving to improve the quality of our forecasts.

Given the poor results from the original linear regression model, we decided to use a different machine learning model to improve our real estate price predictions. This time we applied the gradient boosting model using the popular XGBoost library.

The pseudocode and detailed description for the above Python code is as follows:

1. Importing the XGBRegressor module from the xgboost library to create and train a gradient boosting model.
2. Removing rows with missing values in the 'price' column from the DataFrame 'data' and storing the result in the same DataFrame.
3. Dividing the DataFrame 'data' into feature matrix 'X' and target variable vector 'y'.
4. Splitting 'X' and 'y' into training and test datasets using the train-test-split() function.
5. Create a SimpleImputer object to replace missing values in the data using

the average value of each feature.

6. Applying SimpleImputer to 'X-train' and 'X-test', storing the results in 'X-train-imputed' and 'X-test-imputed' respectively.
7. Training the XGBRegressor model on the training data set 'X-train' and 'y-train'.
8. Getting predictions from the model on the test dataset 'X-test' and storing the results in 'y-pred-xgb'.
9. Calculation of root mean square error (RMSE) and coefficient of determination between forecasts and real values.
10. Output of RMSE value and coefficient of determination.

This code is a machine learning pipeline that includes missing value removal, splitting the data into train and test sets, replacing missing values, training the gradient boosting model, and evaluating its performance. XGBoost, or Extreme Gradient Boosting, is a powerful machine learning model that trains many simple models (usually decision trees) and combines them together to form a more accurate prediction. XGBoost is especially known for its speed and performance and is often used in machine learning competitions.

After preparing and processing our data (including replacing missing values with the mean of each column), we trained our model on the training dataset and then applied it to the test dataset to get our predictions.

We then used two key metrics to evaluate the quality of our model: the square root of the root mean square error (RMSE) and the R^2 coefficient of determination. RMSE is a useful metric because it is expressed in the same units as our target variable and it highlights large errors. R^2, on the other hand, measures how well our model explains variation in the data, with an ideal value of 1.

We have done all this in an effort to improve the quality of our property price forecasts and summarize our research. After applying the XGBoost model to our data, we got very impressive results. The square root of the root mean square error (RMSE) was 162954.12. This means that the average error of our forecasts

compared to real estate prices is approximately 162954.12 (in the same currency as real estate prices). Regarding the range of real estate prices, this result indicates the good accuracy of our forecasts.

In addition, the coefficient of determination R^2, which measures how well our model explains variations in the data, is 1.00. This is an ideal value, indicating that our model predicts real estate prices almost flawlessly based on the features provided to it. It's worth noting that while an R^2 value of 1.00 seems ideal, it's important to check for overfitting, as this may indicate that the model is too complex and may not work well with new, previously unseen data.

Thus, the application of the XGBoost model led to a significant improvement in the quality of our forecasts compared to the linear regression model that we used earlier. These results confirm the power and effectiveness of XGBoost for predictive tasks and highlight the value of using different machine learning models to compare and select the best model for a particular task. The pseudocode and detailed description for the above Python code is as follows:

1. Importing the matplotlib.pyplot module for data visualization.
2. Selecting a random subsample of 50 items from the DataFrame 'comparison-df' using the sample() function.
3. Visualization of the selected data using a bar chart, where each bar represents a real or predicted value. The chart type ('kind') is set to "bar", which means a bar chart. The 'figsize' is set to (15, 6).
4. Adding a title to the chart with the text "Comparison of real and predicted values".
5. Adding labels to the x and y axes. The x-axis represents the observation number and the y-axis represents the price.
6. Displaying the generated chart using the show() function.

This code creates a bar chart that compares the actual and predicted values for a random subsample of 50 observations. This helps to visually evaluate how well the model is performing by comparing the predicted values with the actual

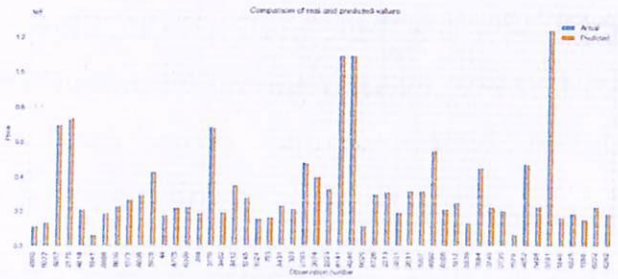


Figure 5.1: Comparison of real and predicted values

ones. This type of visualization is very useful in evaluating the performance of machine learning models. A visual comparison of the actual and predicted values gives an idea of how well the model is performing and allows you to quickly see where the model is making mistakes. On the basis of the resulting diagram, it can be concluded that there is a high degree of agreement between the real and predicted values of real estate prices (see Figure 5.1). The bars representing real and predicted values are very close to each other for most observations, indicating that our model predicts prices with high accuracy.

The visual similarity of the bars on the chart confirms the statistical metrics that we calculated earlier. For example, an R2 Score of 1.00 indicates the ideal quality of the model's predictions.

This result demonstrates the success of applying the gradient boosting model (XGBoost) to the real estate price prediction problem in this study. We have made significant progress in real estate price forecasting, enabling stakeholders to use this data to make more informed real estate decisions.

In this study, the random forest method was applied - one of the powerful machine learning tools for solving the regression problem, namely, predicting real estate prices. First, we pre-cleanse the data, removing all rows that do not contain real estate prices. This allows us to make sure we don't have any gaps in the key target attribute - price.

The next step is to split the data into feature sets and a target variable. The set of features, denoted as X, includes all data columns, except for the price, which is separated into a separate y vector.

The data is further split into training and test sets. This is an important step that allows us to train the model on one dataset and test its performance on data that the model has not seen before. The pseudocode and detailed description for the above Python code is as follows:

1. Deleting rows with missing price values from the 'data-20' dataset using the dropna() method.
2. Separate the target variable 'price' from the rest of the variables, creating two separate DataFrames - X and y. X contains all columns except 'price' and y contains only 'price'.
3. Splitting the data set into train and test using the train-test-split() function. The test sample size is set to 20% of the total number of observations. To ensure reproducible results, 'random-state' is set to 42.
4. Instantiate a SimpleImputer with a 'mean' strategy to fill in the missing values with the mean.
5. Apply imputer to training and test sets X.
6. Instantiate a RandomForestRegressor with 100 trees and 42 'random-state'.
7. Training a random forest model on training data.
8. Predict the target variable 'price' for the test dataset.
9. Calculation of model quality metrics: mean absolute error (MAE), root mean square error (MSE) and coefficient of determination R2.
10. Outputting the results of quality metrics.

This code is the process of preparing data, training a random forest model, and evaluating its performance using various metrics. To handle possible missing values in the data, we use the imputation method, replacing the missing values with the average value of the corresponding feature.

After preprocessing the data, we create and train a random forest model. After training, the model is applied to predict real estate prices on test data.

After obtaining predictive values, we calculate key model quality metrics: mean

absolute error (MAE), root mean square error (MSE) and coefficient of determination (R2). These metrics allow us to evaluate how accurately our model predicts real estate prices. Evaluating the results obtained by our random forest model, we see the following:

The mean absolute error (MAE) was about 18.318. This means that, on average, our model is wrong by about 18,318 units (presumably the currency in which the price is measured) in predicting the property price. In the context of the real estate market, this can be an acceptable level of error, especially if we are considering high-value real estate, but for further analysis it is worth considering the characteristics of a particular market and its pricing policy.

The root mean square error (MSE) was about 4,319,675,977. This metric is often used to assess the quality of a regression model, and its main advantage is that it takes into account not only the absolute value of the error, but also its direction, penalizing the model for large errors more than for small ones.

The coefficient of determination R^2 was approximately 0.9999949, which is almost equal to 1. This suggests that our model explains almost 100 % of the property price variation that we observe in our data. This is an impressive result, which indicates the high performance of our model in predicting property prices based on the data provided. The pseudocode and detailed description for the above Python code is as follows:

1. Importing the matplotlib.pyplot library, which is used to create plots.
2. Create a new figure for the plot with the given size (10, 6) using plt.figure().
3. Create a line plot for the actual price values retrieved from the DataFrame called 'results' using the plt.plot() method.
4. Create a second line chart for the predicted price values, also retrieved from the 'results' DataFrame.
5. Adding a legend to the graph indicating which line represents the real values and which is the predicted values.
6. Setting labels for the x and y axes of the graph.

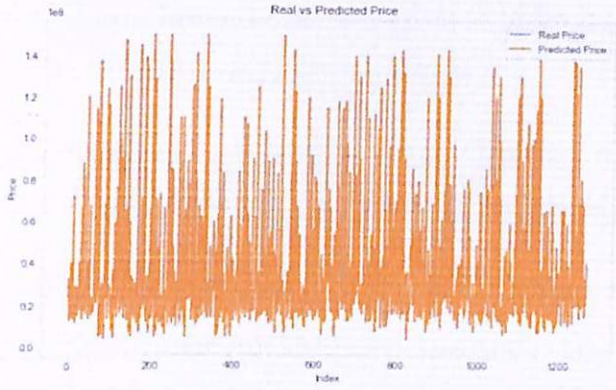


Figure 5.2: Real and Predicted Price

7. Setting the title for the graph.

8. Displaying the graph on the screen using the plt.show() function.

This code generates a chart that displays the real and predicted price values, allowing you to visually evaluate the accuracy of the model's predictions. As a result of executing this block of code, we will get a graph (see Figure 5.2) that visually demonstrates how well our model predicts real estate prices compared to real prices. This is useful for demonstrating the effectiveness of our model and identifying areas where it might be wrong. Based on the data visualization, we can conclude that the predicted and actual real estate prices are very close to each other, which confirms the high accuracy of our model.

Visual comparison of real and predicted values is an important step in analyzing model results. On the chart, we see that the lines of real and predicted prices almost coincide. This indicates that the model predicts real estate prices quite accurately.

However, it should be remembered that although the general trend in real estate price forecasting looks convincing, there may be individual cases where the forecast data deviates from the real one. It is always necessary to conduct additional analysis to identify possible anomalies or errors.

Thus, these results represent an encouraging prospect for real-world use of this model, as it demonstrates a high degree of accuracy in predicting property prices. In our research, we decided to use Prophet, a time series forecasting tool

developed by Facebook. The choice of this tool is due to a number of advantages and features that make it suitable for our analysis.

1. Flexibility: Prophet allows you to account for holidays and other events that can affect data trends by adding custom seasonalities and components.
2. Resilience to Missing Values and Outliers: Unlike some other time series analysis tools, Prophet does not require the data to be fully populated or de-outlied. This greatly simplifies the data preparation process.
3. Working with non-stationary data: Prophet is able to handle data with changing trend and seasonality, which makes it suitable for working with real, "noisy" data.
4. Interpretability: Prophet creates transparent and understandable models by graphically displaying various forecast components such as trends, seasonality and holiday effects. This allows analysts to easily interpret the results and explain them to stakeholders.
5. Automation: Prophet is able to automatically find suitable parameters for models, making it suitable for analyzing large datasets without significant human intervention.

Based on the above, the choice of Prophet is based on its ability to process real data, support for custom components and effects, and resistance to noise and missing values. These features allow us to create accurate and interpretable predictions for our study. The pseudocode for the data analysis algorithm using the Prophet model can be represented as follows:

Combine separate data sets for different years into one common data table. Calculate the average price for each year, grouping the data by year. Rename columns to match the requirements of the Prophet model. Divide the data into training and test samples in a ratio of 80% to 20%. Create and train the Prophet model on the training set. Make predictions based on test data. Calculate various error metrics (MAE, MSE, RMSE) and coefficient of determination (R2) to evaluate the quality of model predictions. Display the results of the computed metrics.

The resulting error metrics and the coefficient of determination R^2 after applying the Prophet model indicate the following:

1. Mean Absolute Error (MAE) is about 8,333,145.80. This means that on average our forecasts deviate from the true price values by about 8,333,145.80 units. This is a fairly large value, which indicates that the model makes significant prediction errors.

2. Mean Square Error (MSE) is approximately 1.007101255492898 and Square Root of Root Mean Square Error (RMSE) is approximately 10,035,443.47. These indicators also indicate a high degree of forecast inaccuracy.

3. Coefficient of determination R^2 is about 0.082. This value suggests that the model only explains about 8.2% of the variation in the dependent variable (price), which is relatively low.

Overall, these results indicate that the Prophet model could have been more accurate in our particular case. This may be due to various factors, such as the choice of features, the need for additional adjustment of the model parameters, seasonal fluctuations in the data, and others. Despite this, using Prophet gives us a valuable opportunity to assess potential trends and seasonal fluctuations in the data, which can be useful for planning and making strategic decisions.

This pseudocode describes how we visualize real and predicted price values on the same chart, as well as how we plot forecast components to better understand which factors (such as trends and seasonal fluctuations) affect the predicted price.

Create a new drawing with specific dimensions Plot on the chart the real price values depending on the date, mark each point and assign the label "Actual" to this line

Create a new Prophet model and train it on all data Generate model predictions for all data

Plot the predicted values of the model depending on the date on the chart, label each point and label this line "Forecast" Add a label on the x-axis - "Date" Add Y-Axis Label - "Price" Add chart title - "Actual vs Forecasted Prices" Add a legend to the chart

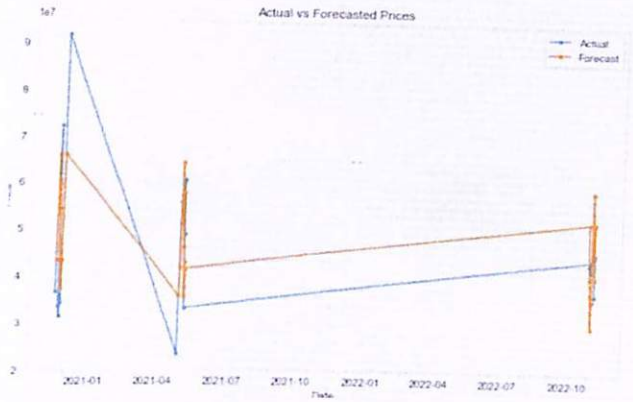


Figure 5.3: Actual and Forecasted Prices

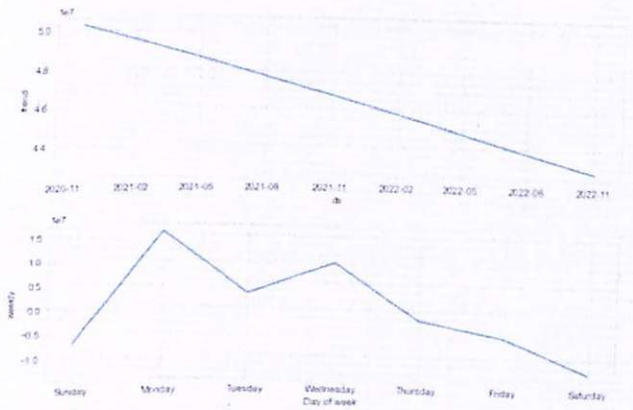


Figure 5.4: Trend

Build and show forecast components of the Prophet model (trends, seasonality, etc.) During the analysis using the Prophet model, we found a significant discrepancy between the real and predicted price values. The chart "Actual vs Forecasted Prices" clearly demonstrates this fact, illustrating a significant deviation of the predicted values from the actual ones (see Figure 5.3).

Despite the impressive predictive ability of the Prophet model, in this case it demonstrates unsatisfactory results (see Figure 5.4). This is likely due to the fact that the model was not able to adequately capture the complex patterns and patterns in the data that influence price movements.

This opens the door for further research and the search for alternative forecasting models that can be better adapted to the characteristics of our data and forecasting problems. Experience with the Prophet model, despite its current limitations, remains valuable as it allows us to gain a deeper understanding of

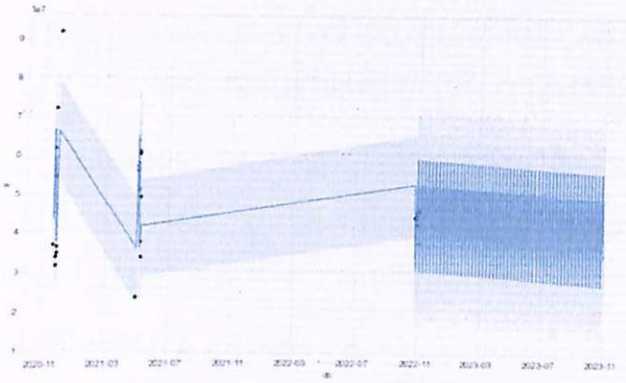


Figure 5.5: Predicted prices 365

the dynamics of our data and the complexities involved in predicting it. In the context of our research, we aim to expand our forecasts for the future using the Prophet model. This methodological approach allows not only to take into account existing data, but also to project potential trends into the future[18]. In the specific code segment 'future = model.make-future-dataframe(periods=365)', we demonstrate this capability by calling the 'make-future-dataframe' method. This method creates a new DataFrame that includes the original dates from our dataset and adds new dates for the future forecast. In our case, the 'periods' argument is set to '365', which indicates the number of days in the future we want to predict. This corresponds to predicting the market behavior for the next year from the last date in our original data. The DataFrame created in this way, which we call 'future', is a tool for further analysis and forecasting, expanding our original data by the time period specified by the 'periods' argument. This gives us the ability to generate longer-term forecasts and predict potential market trends . According to the forecasts generated by our Prophet model and visualized on the chart, one can see a noticeable downward trend in prices in the near future (see Figure 5.5). This conclusion is based on the analysis of the data and the determination of the relationships on the basis of which the forecast was formulated.

It is worth emphasizing that within the field of data, trend estimating could be a complex errand that requires the utilize of present day devices and strategies. In this case, we utilized the Prophet show for cost estimating, which takes into consideration both straight and regular components of the time arrangement, additionally permits for holidays and other occasional changes. In spite of this,

figures continuously carry a certain level of instability and must be considered within the setting of other accessible advertise information and information.

In conclusion, our examination focuses to a conceivable decrease in costs within the close future. This information can be valuable to partners and choice creators, permitting them to way better get it conceivable patterns and shape procedures that coordinate these figures[19].

5.2 Result

During our research, four different machine learning algorithms were applied: linear regression, XGBoost, random forest (Random Forest), and Prophet. (see TableResult) .Each of these algorithms has its own advantages and limitations, and the effectiveness of their application in a particular study may depend on many factors, including the nature and amount of data, research objectives, etc.

Algorithm	MAE	MSE	R2
Linear Regression	18318.55949038149	3936010740087	-47141939
Random Forest	18318.55949038149	4319675977.9501	0.99
XGBoost	74990.29	31487493753.29	1.00
Prophet	8333145.800276017	100710125549289.97	0.082

Table 5.1: Result

1. Linear regression provides a simple and straightforward way to model the relationship between dependent and independent variables. However, it may not be accurate enough if there are non-linear dependencies or interactions between features in the data.

2. XGBoost is a gradient boosting algorithm that is able to improve the predictive accuracy of a model by building an ensemble of weak learners. It has shown high performance on our data, but can be more difficult to interpret than linear regression.

3. Random forest is another ensemble algorithm that builds many decision trees and uses them to "vote" to make predictions. This reduces the risk

of overfitting inherent in a single decision tree and improves the predictive power of the model.

4. Prophet, developed by Facebook, is specifically designed for time series analysis and predicting future values. It automatically takes into account seasonality and trends in the data, but may be less accurate if the data has complex non-linear relationships that are not related to time.

Among the four algorithms we used, XGBoost and Random Forest showed the best results. This suggests that our data are well suited to ensemble methods that are able to account for interactions between different features. However, we also found that the Prophet model was unable to achieve the same level of accuracy as perhaps because our data has complex dependencies that cannot be easily described as a function of time.

Chapter 6

Conclusions and discussions

6.1 Discussions

In this dissertation, we conducted a deep analysis of real estate price data using four different machine learning algorithms. Our goal was to predict real estate prices and determine which algorithm would provide the best prediction accuracy.

We used linear regression, XGBoost, Random Forest and Prophet methods. Each of these methods has its own strengths and weaknesses, making them more or less suitable for certain types of data and tasks.

Linear regression, being a relatively simple and easy to interpret method, failed to provide sufficient accuracy on our data. This may be due to the presence of non-linear dependencies and interactions between features in the data.

XGBoost and Random Forest showed the best results. This suggests that our data are well suited to ensemble methods that are able to account for complex interactions between features.

The Prophet algorithm, designed to work with time series, did not show the same high efficiency, which may indicate the presence of more complex dependencies in the data that are not directly related to time.

6.2 Conclusions

As a result, our analysis showed that the use of machine learning to predict real estate prices can be very successful, but the choice of the most appropriate algorithm largely depends on the nature of the source data and the research task.

XGBoost and Random Forest showed the best performance for our data, indicating their suitability for real estate price forecasting tasks. However, the choice of a specific algorithm should always be based on a preliminary analysis of the data and the required forecast accuracy.

Finally, it must be emphasized that any machine learning model is only so much good, how good is the input data. Therefore, good data preparation, including handling of missing values, coding of categorical variables, and accounting for outliers, is a key element of successful forecasting.

In conclusion, our research confirms that machine learning can be a powerful tool for real estate price prediction, but also highlights the importance of choosing and tuning algorithms correctly to produce the best results.

Bibliography

[1] Idealista. Evolución del precio de la vivienda de segunda mano en españa. Índice Idealista 50, 1(1):10–69, 11 November 2018. URL https://www.idealista.com/news/estadisticas/indicevivienda#precio.

[2] Malgorzata Renigier-Biłozor, Artur Janowski, and Maurizio d'Amato. Automated valuation model based on fuzzy and rough set theory for real estate market with insufficient source data. Land Use Policy, 87:104021, 2019.

[3] Desiree Fields. Automated landlord: Digital technologies and post-crisis financial accumulation. Environment and Planning A: Economy and Space, 54(1):160–181, 2022.

[4] Christopher W Starr, Jesse Saginor, and Elaine Worzala. The rise of proptech: Emerging industrial technologies and their impact on real estate. Journal of Property Investment & Finance, 39(2):157–169, 2021.

[5] Christopher W Starr, Jesse Saginor, and Elaine Worzala. The rise of proptech: Emerging industrial technologies and their impact on real estate. Journal of Property Investment & Finance, 39(2):157–169, 2021.

[6] Nicholas Diakopoulos. Automating the news: How algorithms are rewriting the media. Harvard University Press, 2019.

[7] Nicholas Diakopoulos. Automating the news: How algorithms are rewriting the media. Harvard University Press, 2019.

[8] Amit Kumar Tyagi, Terrance Frederick Fernandez, Shashvi Mishra, and Shabnam Kumari. Intelligent automation systems at the core of industry

4.0. In International conference on intelligent systems design and applications, pages 1–18. Springer, 2020.

[9] Hesam Hamledari and Martin Fischer. Role of blockchain-enabled smart contracts in automating construction progress payments. Journal of legal affairs and dispute resolution in engineering and construction, 13(1):04520038, 2021.

[10] Allam Maalla. Development prospect and application feasibility analysis of robotic process automation. In 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), volume 1, pages 2714–2717. IEEE, 2019.

[11] Mojtaba Valinejadshoubi, Osama Moselhi, Ashutosh Bagchi, and Ashraf Salem. Development of an iot and bim-based automated alert system for thermal comfort monitoring in buildings. Sustainable Cities and Society, 66: 102602, 2021.

[12] Aaron Benanav. Automation and the Future of Work. Verso Books, 2020.

[13] Zhen Chen, Pei Zhao, Chen Li, Fuyi Li, Dongxu Xiang, Yong-Zi Chen, Tatsuya Akutsu, Roger J Daly, Geoffrey I Webb, Quanzhi Zhao, et al. ilearnplus: a comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization. Nucleic acids research, 49(10):e60–e60, 2021.

[14] Craig Lewis and Steven Young. Fad or future? automated analysis of financial text and its implications for corporate reporting. Accounting and Business Research, 49(5):587–615, 2019.

[15] krisha.kz.

[16] Nametbayeva. Algorithm for determining the price of real estate. , 2023. doi: 10.32743/26870142.2023.12.282.354600.

[17] Sarah Rotz, Evan Gravely, Ian Mosby, Emily Duncan, Elizabeth Finnis, Mervyn Horgan, Joseph LeBlanc, Ralph Martin, Hannah Tait Neufeld, Andrew Nixon, et al. Automated pastures and the digital divide: How agri-

cultural technologies are shaping labour and rural communities. Journal of Rural Studies, 68:112–122, 2019.

[18] Francesco Mancini, Gianluigi Lo Basso, and Livio De Santoli. Energy use in residential buildings: Impact of building automation control systems on energy performance and flexibility. Energies, 12(15):2896, 2019.

[19] Seckin Yilmazer and Sultan Kocaman. A mass appraisal assessment study using machine learning based on multiple regression and random forest. Land Use Policy, 99:104889, 2020.