Information system for real estate trading operations based on the data analysis

Oleh Veres[1,†], Pavlo Ilchuk[1,†] and Olha Kots[1,*,†]

[1] Lviv Polytechnic National University, Stepana Bandery str. 12, Lviv, 79013, Ukraine

Abstract

The process of buying and selling property heavily depends on accurate real estate valuation. We have carefully studied the existing programs with the help of which we carry out real estate transactions, described their features, advantages and disadvantages. Traditionally, real estate valuation relied mainly on manual data analysis and subjective estimates, often resulting in mistakes and delays. Implementing machine learning algorithms has shown to be effective in addressing this issue, offering several benefits over manual assessments: high accuracy, elimination of subjectivity and bias, time efficiency, cost reduction, utilization of geospatial data, and well-supported results. The process of creating a machine learning model is conventionally divided into four stages. Linear regression, decision tree, nearest neighbor, support vector, and random forest algorithms were tested using standard parameters. The R-squared coefficient of determination was chosen as the main metric. After comparing the coefficient of determination of the results, it became clear that the "random forest" algorithm showed the best result. Using manual hyperparameter selection for this algorithm, the mean absolute error of the predicted value is 8.49%, with a median error of 1.9%. The built model meets the established quality requirements and is ready for implementation in the information system for forecasting the value of real estate. The system was divided into three separate services, each responsible for a specific set of functions. The purpose of each service is outlined, along with their main functions and the connections between them. Modular and end-to-end testing of the server and user parts of the program was conducted to confirm the readiness of the system for use. All services function properly and interact seamlessly with each other.

Keywords

Data Analysis, Machine Learning, Real Estate, Trading, Evaluation, Information System

1. Introduction

Real estate transactions involve significant amounts of money, and thading decisions need to be made on the basis of relevant data. Correct valuation of real estate plays a crucial role in the trading process. The value is determined by various factors, such as location, kitchen and room area, condition, year of construction, amenities, nearby infrastructure, neighborhood development trends, market trends, and many others. Overpriced properties

---

MoDaST-2024: 6th International Workshop on Modern Data Science Technologies, May, 31 - June, 1, 2024, Lviv-Shatsk, Ukraine

* Corresponding author.

† These authors contributed equally.

oleh.m.veres@lpnu.ua (O. Veres); pavlo.g.ilchuk@lpnu.ua (P. Ilchuk); olha.o.kots@lpnu.ua (O. Kots)

0000-0001-9149-4752 (O. Veres); 0000-0003-4636-2309 (P. Ilchuk); 0000-0001-7123-3635 (O. Kots)

can linger on the market without attracting buyers, while undervalued properties can result in substantial losses for the seller. Accurate valuation is crucial for making informed decisions and preventing financial losses [1].

However, the cost of hiring specialists for evaluating real property can be substantial. Additionally, human limitations should be considered, especially when dealing with large volumes of data. It's important to remember that the real estate search process can be time-consuming, and sometimes the need to find a property can be urgent. Therefore, the task of developing an information system that will help meet the needs of both sellers and buyers is urgent.

The implementation of this system will simplify and speed up the process of finding the best property for purchase, as well as the process of its valuation for further sale. Real estate sellers will be able to quickly assess the value of their property depending on the parameters, location, and current market conditions, and buyers will be able to receive a list of recommended properties that are priced according to their real characteristics or below market value.

Traditionally, real estate valuation has relied heavily on manual data analysis and subjective estimates, frequently leading to inaccuracies and delays. Therefore, the use of machine learning methods to predict the value of real estate objects is relevant, as it will ensure transparency in the real estate market, reduce the cost of realtors and real estate agents, and enable users to make more efficient decisions regarding the purchase or sale.

2. Analysis of recent research and publications

Utilizing machine learning to forecast real estate values has introduced unparalleled levels of precision, productivity, and transparency. There's no longer a necessity to solely depend on the expertise of industry professionals, who often grappled with vast data sets and intricate criteria when establishing real estate prices.

The real estate market dynamics demand real-time adaptability, a task at which artificial intelligence excels. Conventional valuation methods frequently rely on outdated data and struggle to accommodate rapidly changing market conditions [5, 6]. Conversely, artificial intelligence effortlessly integrates real-time changes, giving it a distinct advantage [2].

Consequently, the value estimates generated not only draw from historical data but also accurately reflect the present market conditions, aiding stakeholders in comprehending the constantly evolving landscape. The ability for real-time adaptation distinguishes artificial intelligence from traditional approaches and highlights its crucial role in navigating a rapidly evolving market, showcasing its revolutionary potential [3].

3. Benefits of employing artificial intelligence in real estate valuation

Enhanced Precision. The use of artificial intelligence tools helps to avoid subjectivity and bias in determining the value of real estate.

Time Efficiency. Time is a precious resource in property valuation. Traditional property valuation methods often demand days or weeks to produce comprehensive assessments, whereas AI-driven processes excel in efficiency, yielding results swiftly. The time required for real estate valuation is significantly reduced when data is analyzed by an AI system. This accelerates transactions, introduces flexibility into the valuation process, and enables stakeholders to promptly address shifting market dynamics.

Cost Reduction. Artificial intelligence's efficacy also impacts the financial aspect of property valuation. Leveraging machine learning techniques diminishes the necessity for repeated valuations by efficiently identifying trends from various data sources and synthesizing them. This significantly reduces costs for both buyers and sellers.

Integration of Geodata. Geospatial data plays a crucial role in real estate valuation, encompassing factors such as a property's proximity to amenities, location within flood zones, or adjacency to industrial areas. Artificial intelligence algorithms can seamlessly incorporate geographic information systems to include these variables in real-time valuations, providing a level of detail that was previously challenging to attain. [4].

Clear understanding. The openness of valuations ensures transparency in real estate transactions. The algorithms employed in AI-based valuations are not "black boxes"; they provide clear justifications for the conclusions. Stakeholders are provided with a detailed explanation of each valuation methodology. This transparency fosters trust and facilitates well-informed decision-making.

Enhanced investment decisions. The empirical foundation of AI-powered valuation furnishes stakeholders with a potent tool to refine investment choices. Investors, buyers, and sellers leverage the resultant real estate valuation data to make more informed decisions. The accuracy of valuation provided by artificial intelligence mitigates risks in determining expected returns on real estate and analyzing market trends for strategic decision-making. This approach prevents the potential of overpaying for a purchase or underestimating the value of real estate, resulting in more successful and informed outcomes [2].

3.1. Comparison of current systems for real estate sales, purchases, and leases

The study focused on analyzing the most popular applications for real estate sales, purchases, and rentals in the Ukrainian market, specifically DIM.RIA [7] and OLX [8]. These services boast a considerable advantage due to their popularity in Ukraine, resulting in vast and expanding repositories of real estate data. Leveraging this data in service development is feasible, as it can be acquired internally. Having thoroughly researched the most popular applications used abroad (Zillow [9], Realtor [10], Redfin [11], PropStream [12]), the comparative table (Table 1) is provided to emphasize their capabilities and compare them with the system under development.

Of all the services, PropStream is the most similar in terms of functionality, as it provides a wide range of data analysis features. However, it has a complicated user interface.

The key distinction between the system under development and existing solutions lies in the approach to real estate valuation. While existing solutions primarily rely on basic physical property characteristics (such as area, number of rooms, floor, year of construction, availability of parking spaces, etc.), the system being developed will also factor in location.

Table 1
Comparative analysis of the system with competitive applications

Features	Applications						
	OLX	DIM.RIA	Zillow	Realtor	Redfin	PropStream	System under development
Evaluation of real estate by its parameters for the seller	-	+	+	+ (paid)	+	+	+
Evaluation of real estate by its parameters for the buyer	-	Partly	+	+	-	+	+
Real estate inspection service (condition and compliance with the ad)	-	+ (paid)	+	+	+	+	-
Real estate offer service according to selected parameters	-	+ (paid)	+	+	+	+	+
Analytical report service for real estate in the selected region	-	+ (paid)	-	-	+	+	-
Service of automatic notification of new ads by selected filters	-	+	+	+	+	+	+
Reflection of factors that positively and negatively affect the value of real estate	-	-	Partly	Partly	Partly	+	+
Displaying important locations and infrastructure near the selected property	-	+	+	+	+	+	+
Simple and intuitive user interface	+	+	+	+	+	-	+
Availability of analytical tools for investors	-	-	-	-	-	+	Partly
Displaying useful data about the area where the property is located	-	+	+	+	+	+	+

The system will analyze the nearest and most significant locations within a radius of 500, 1000, and 3000 meters from the property, considering how they may positively or negatively impact its value. Emphasis on geospatial data is one of the most important factors in real estate valuation [4].

Existing applications in the Ukrainian market fall short of meeting all user needs, lacking sufficient functionality for recommendations and analytics. Hence, it is imperative to develop a new service that addresses these deficiencies, enabling users to make more informed decisions regarding real estate transactions.

4. Creating a conceptual model

4.1. Modeling business processes and project requirements

The challenge in predicting real estate values stems from the scarcity of resources, particularly the confidentiality surrounding real estate data. The current approach to real estate transactions in Ukraine is complicated and even somewhat outdated [13, 14]. The main problem is that buyers and sellers can choose one of two options.

The first is to do everything yourself. Stakeholders have to analyze the market on their own, compare many properties with each other, determine the feasibility of buying or the fair price for selling, while putting in a lot of effort and spending a lot of time.

The second approach is to delegate the task by using the services of appraisers and real estate experts. This will greatly reduce the amount of effort and time spent and will likely lead to a better result. However, such services are usually expensive.

The process of buying real estate is quite complex and potentially requires the involvement of third parties. In this case, the system for buying and selling real estate serves only as a data filtering tool and does not perform recommendation functions.

It should also be paid the attention to how long the process of searching for and evaluating real estate takes, if you use the services of a realtor. If the buyer does not find an option that meets his or her requirements and meets the expected budget, he or she will have to contact the realtor again to defend his or her interests and explain the requirements. And the realtor, in turn, should try to adapt to the client's requirements, while looking for possible alternatives, each time assessing the correspondence between the ad and the property value.

The "to-be" business process diagram (Figure 1), which includes the use of the information system, shows that the decision-making process for purchasing real estate has been greatly simplified and involves far fewer steps. After filtering the property by the specified parameters, the buyer can choose the options he is interested in.

Thanks to the system, the user can do without the services of a real estate appraiser, so he does not need to explain his needs, requirements and limitations to outsiders. This is beneficial not only from an economic point of view, but also from a practical point of view, since the buyer knows his own needs best. It is also important to note that in case of a mismatch between desires and capabilities, it is easier for the buyer to compromise with his requirements than if other people demanded it from him.

Now let’s draw up requirements for the developed system.

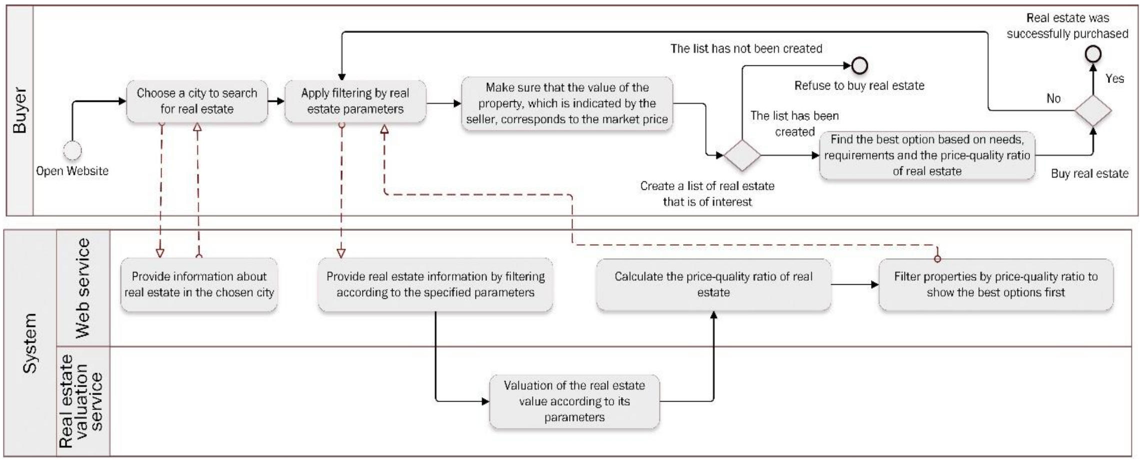

Figure 1: The "to-be" business process diagram

Business requirements. This information system aims to streamline and expedite the process of identifying optimal real estate for purchase and evaluating its value for subsequent sale. The implementation of the system in the real estate sector will be advantageous for buyers, sellers, agents, and investors alike, streamlining their business processes.

The implementation of the system yields several positive effects:

- Enhanced real estate search: the system will enable buyers to swiftly and efficiently discover properties that align with their requirements and financial capacities.
- Enhanced valuation accuracy: real estate sellers will have the capability to more precisely determine the value of their property, facilitating the identification of buyers and the negotiation of profitable deals.
- Reduced investment risk: Investors will utilize the system to analyze and select properties with greater potential, minimizing their risk exposure.
- Positive impact on the real estate market: all stakeholders (buyers, sellers, agents and investors) will receive tools that will help improve the efficiency of the real estate market and facilitate interaction between them.

User requirements. The information system should provide for several users with different roles, which, accordingly, will have a different list of capabilities and functions that they can operate.

List of functions for unauthorized users:

1. Possibility of logging and registration.
2. Ability to view the property for sale in a grid view and on a map.
3. Ability to filter data depending on the specified parameters.
4. View detailed information about the selected property.

Unique features for authorized users:

1. All available unauthorized user functions.
2. The ability to enter data about your property.
3. Determine the value of the property and the cost of rent, depending on the input parameters.
4. Ability to see the factors that most affect the valuation of your own property (negative and positive).
5. The ability to view the user's real estate for sale in your personal account.
6. Ability to view real estate for sale by other users in the form of a table and on a map.
7. Ability to filter data depending on the specified parameters.
8. View the factors that have the greatest impact on the valuation of the selected property (negative and positive).
9. Receive recommendations in your personal account, depending on the predefined parameters.
10. Receive notifications in case of new recommendations for purchase.

Unique features for the user with the investor role:

1. All available functions of the authorized user.
2. Ability to view graphs of changes in property value over time.
3. Ability to receive analytical data at the level of regions, cities and city districts
4. Availability of a calculator that predicts the time required to pay off the property.
5. The ability to receive notifications in case of the appearance of recommended real estate according to the specified filters.

It is also needed to implement an administration page. A user with administrator rights should be able to change the roles of existing users, as well as add, edit, and delete all existing system object entities.

Functional requirements. Below is a list of actions performed by this system:

- To access the system's functionality, users must be able to register and log in.
- The system should allow users to search for available properties based on various criteria, such as price, location, property type and other parameters.
- The system should provide recommendations for real estate that meets the users' requirements and criteria.
- The system should help users determine the value of real estate based on real data and market analysis.

The objects of actions in such a system are system users: unauthorized user, authorized user, investor and administrator, as well as real estate objects.

The main types of actions are: searching for real estate, filtering results, receiving recommendations based on user-specified requirements, estimating the value, and displaying the parameters that most influenced the value in the evaluation process.

Systems can be divided into interactive and automated according to the nature of their operation. Interactive systems allow users to interact with the system by entering search criteria, viewing recommendations, and analyzing the results. Automated systems are those that analyze data and provide recommendations and cost estimates based on algorithms and data.

Non-functional requirements. So, first, it is needed to define the business rules for using the system:

- The system must comply with legal requirements and standards in the field of real estate and data protection.
- The system must contain a link to the original source of the ad.
- The system should provide users with objective and accurate recommendations on real estate and its value.

The next step is to list the most important quality attributes that need to be ensured for correct operation of the system:

- Efficiency - the system must work quickly and efficiently, even with a large volume of data and requests.
- Flexibility - the system should be open to expansion to allow for future implementation of new features and capabilities.
- Scalability - the system must be able to scale to handle growing volumes of data and users.
- Security - the system must provide an adequate level of protection for user data, including confidential real estate information and personal data.
- Reliability - the ability to ensure uninterrupted operation of the system and the ability to recover in the event of errors or failures.
- Convenience - the system should have a simple and intuitive user interface that will allow you to use all the planned functions.
- Mobility - the system should be available on different devices and platforms.

To be able to realize high efficiency and scalability of the system, it is necessary to consider the possibility of applying containerization of the system and the use of orchestration tools. Although the system is not designed to be used by a large number of users, it should have the potential for widespread use.

Another important component is the use of data caching functions, which will allow you to efficiently process the same user requests, which are expected to be quite large.

4.2. Modeling objects within the subject area

Based on the program's tasks and functions, as well as the intended users and their capabilities, a use case diagram was constructed (Figure 2). This diagram visualizes the user and functional requirements of the system.

The diagram depicts four system actors: user (authorized), unauthorized user, administrator and investor. Users can potentially elevate their role within the hierarchy to access a broader range of functions if needed.

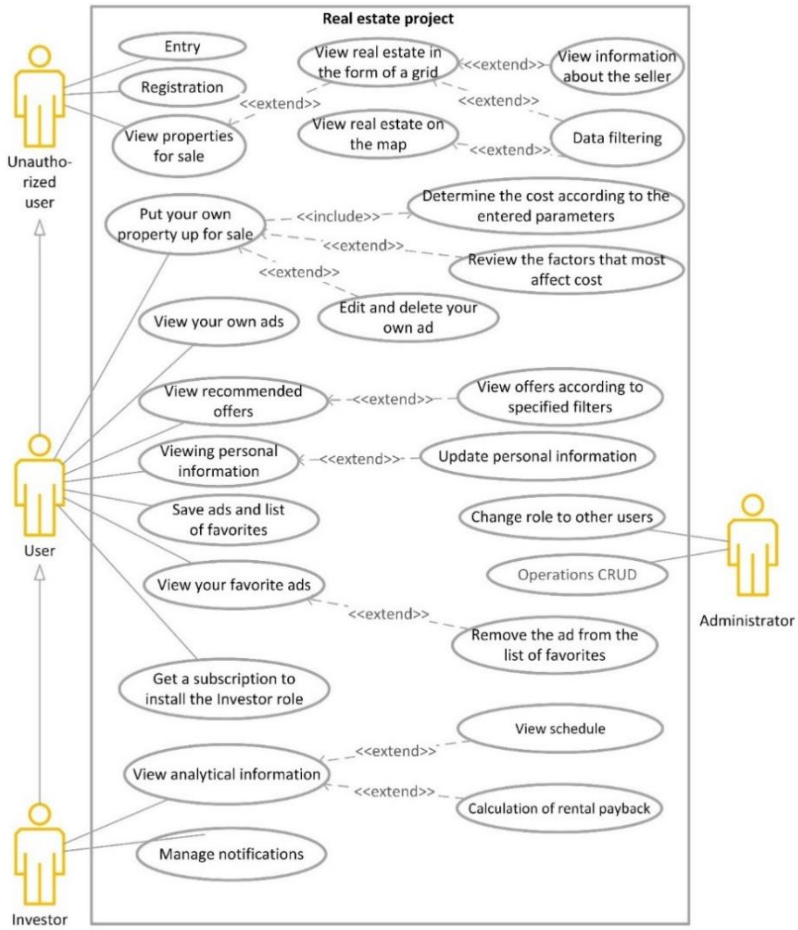

Figure 2: Use case diagram

User (or authorized user) – possesses the same functions as an unauthorized user but with significantly expanded options. He has the ability to create, edit, and/or delete their own real estate advertisements.

Unauthorized user – has the ability to view real estate for sale in both grid and map formats, utilizing filtering tools by parameters. Additionally, they can access comprehensive information about the real estate unit, including the contact details of the advertisement's author.

Administrator – is tasked with managing all system entities, including the ability to modify user roles. Upon completing the definition of system requirements, user roles, functions, and capabilities, the next stage involves detailing the program implementation. During this stage, the entities available in the system, along with their methods, attributes, and relationships with each other, will be determined.

Investor – is a unique system role that extends the capabilities of an authorized user. It provides access to analytics tools, allowing investors to view graphs, estimate the cost of renting a real estate unit, and calculate the payback period through rent. Additionally, investors can opt to receive notifications about recommended real estate according to specified filters.

Let's build a class diagram (Figure 3) for the part of the software system responsible for searching, filtering, modifying and analyzing data, as well as for training machine learning models, testing them and applying them to solve real estate valuation problems.

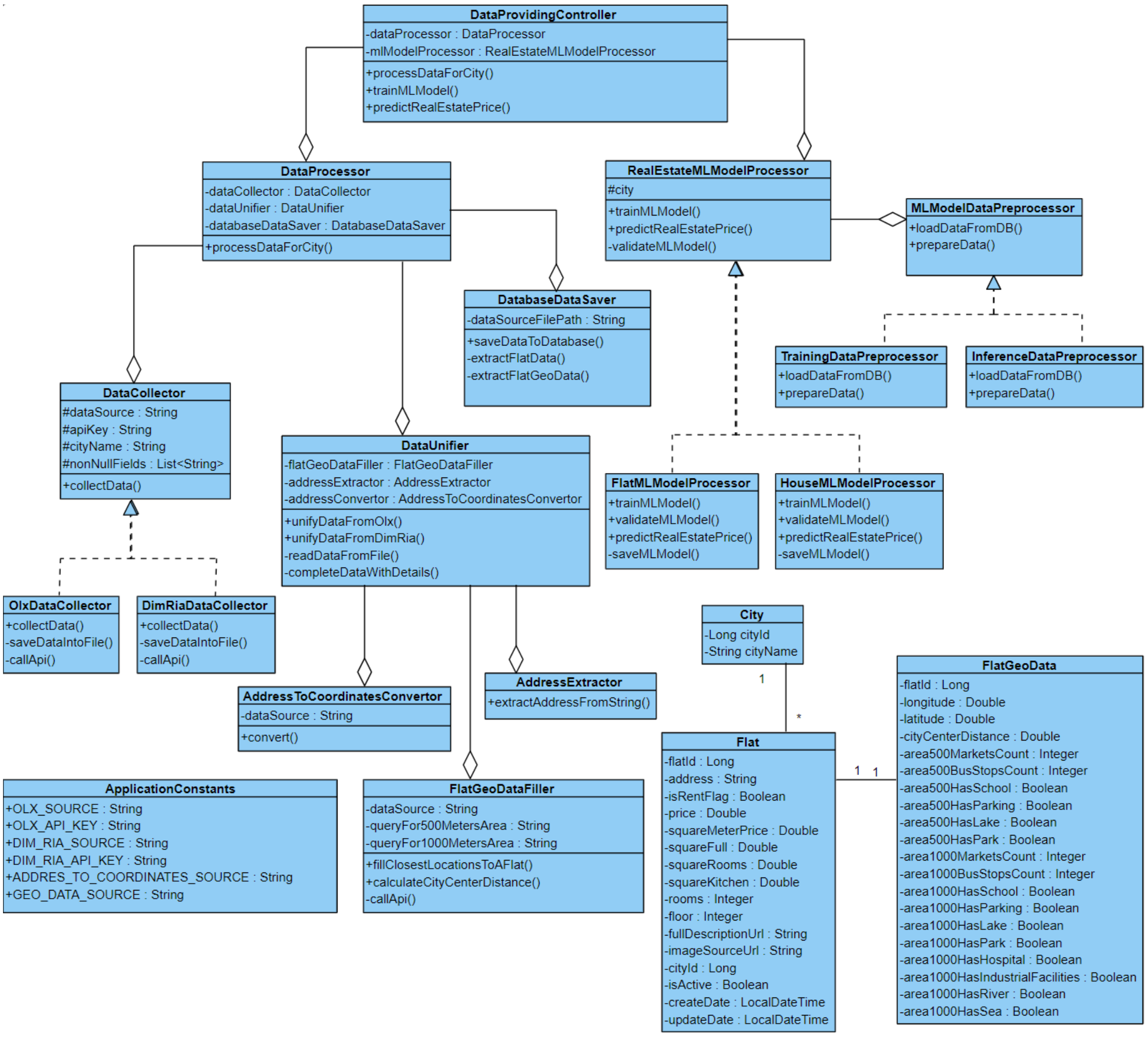

Figure 3: Class diagram

The DataProvidingController class. Receiving and processing HTTP requests from users. *Class attributes*: dataProcessor - an object of the DataProcessor class responsible for working with data; mlModelProcessor - an object of the MlModelProcessor class responsible for working with machine learning models. *Class methods*: processDataForCity - download, process, and save data related to real estate for a given city; trainMLModel - train a machine learning model based on the data available in the system; predictRealEstatePrice - predict the price of real estate transferred as a parameter based on the machine learning model available in the system.

The DataProcessor class. Collection, processing, unification and storage of data related to real estate. *Class attributes*: dataCollector - an object of the DataCollector class responsible for collecting real estate data; dataUnifier - an object of the DataUnifier class responsible for unifying data; databaseDataSaver - an object of the DatabaseDataSaver class responsible for saving data. *Class methods*: processDataForCity - collect real estate data from various sources, process it, unify it and save it to the database.

The DataCollector class. An abstract class that is responsible for collecting real estate data for a given city from a specific information source using specified filters. *Class attributes*: dataSource - source of information or link to the resource; apiKey - key to use the resource; cityName - name of the city to search for data; nonNullFields - list of required fields for filtering. *Class methods*: collectData - an abstract method for collecting data from a specific resource.

The OlxDataCollector class. Implementation of the abstract DataCollector class, which is responsible for collecting real estate data from the OLX resource. It uses the data access key and filters the data by city name and required fields. *Class methods*: collectData - collecting real estate data from the OLX resource; saveDataIntoFile - a private method for saving the received data as a .csv file; callApi - a private method for calling the OLX service API (application program interface) to receive data by specified filters.

The DimRiaDataCollector class. Implementation of the abstract DataCollector class, which is responsible for collecting real estate data from the DimRia resource. It uses the data access key and filters the data by city name and required fields. *Class methods*: collectData - collects real estate data from the DimRia resource; saveDataIntoFile - a private method for saving the received data as a .csv file; callApi - a private method for calling the DimRia service API (application program interface) to receive data by specified filters.

The DataUnifier class. Unification of data from different resources into a single format. Also responsible for modifying and supplementing data related to the location of real estate. *Class attributes*: addressExtractor - an object of the AddressExtractor class that is responsible for converting the address of the property into a single format used in this system; addressConvertor - an object of the AddressToCoordinatesConvertor class that is responsible for obtaining geographic coordinates by the full address; geoDataFiller - an object of the GeoDataFiller class that is responsible for searching and filling in data on geolocation near a specific property. *Class methods*: unifyDataFromOlx - unification of data received from the OLX resource; unifyDataFromDimRia - unification of data received from the DimRia resource; readDataFromFile - a private method for reading data from a given file; completeDataWithDetails - a private method for supplementing real estate data with geodata.

Class AddressExtractor. Extract the address and bring it to a single format used in the application. *Class methods*: extractAddressFromString - extract the address from the text and bring it to a single format.

Class AddressToCoordinatesConvertor. Getting geographic coordinates of a location by its full address using a third-party service. *Class attributes*: dataSource - a link to a third-party API that converts the address to coordinates. *Class methods*: convert - getting the coordinates of the location by its address.

The GeoDataFiller class. Search and fill in geolocation data near a specific property. *Class attributes*: dataSource - a link to a third-party API that returns a list of locations near the specified geographic coordinates; queryFor500MetersArea - a request to the API with the search filter applied to search for locations within a radius of 500 meters; queryFor1000MetersArea - a request to the API with the search filter applied to search for locations within a radius of 1000 meters. *Class methods*: fillClosestLocationsToRealEstate - search and fill in data on real estate objects with geodata; calculateCityCenterDistance - calculate the distance from a given real estate location to the city center; callAri - a private method for calling the API of a third-party service to search for locations near a given location, using filters.

The DatabaseDataSaver class. Saving data from a .csv file to a database. Convert file rows to entities of different tables. *Class attributes*: dataSourceFilePath - the full path of the .csv file in the system. *Class methods*: saveDataToDatabase - saves data from a .cv file to a database; extractFlatData - a private method that converts a .cv file string to the Flat property table entity; extractFlatGeoData - a private method that converts a .csv file string to the FlatGeoData property geodata table entity.

The RealEstateMLModelProcessor class. An abstract class for training and validating a machine learning model, as well as predicting the value of a transferred real estate object. *Class attributes*: city - the name of the city for which you want to train the machine learning model. *Class methods*: trainMLModel - train the machine learning model based on the data in the database; predictRealEstatePrice - predict the value of the transferred real estate using the model; validateMLModel - validate the machine learning model.

The FlatMLModelProcessor class. Implementation of the abstract class RealEstateMLModelProcessor to train and validate a machine learning model for the Flat object, as well as to predict the value of the transferred real estate object. *Class methods*: trainMLModel - train the machine learning model based on the data in the database; validateMLModel - predict the value of the transferred real estate using the model; predictRealEstatePrice - validate the machine learning model; saveMLModel - save the trained model.

The HouseMLModelProcessor class. Implementation of the abstract class RealEstateMLModelProcessor for training and validation of the machine learning model for the House object, as well as prediction of the value of the transferred real estate object. *Class methods*: trainMLModel - train the machine learning model based on the data in the database; validateMLModel - predict the value of the transferred real estate using the model; predictRealEstatePrice - validate the machine learning model; saveMLModel - save the trained model.

The MLModelDataPreprocessor class. An abstract class responsible for processing data used in machine learning models. *Class methods*: loadDataFromDB - load data from the database; prepareData - prepare data before using it in the model.

The TrainingDataPreprocessor class. Implementation of the abstract class MLModelDataPreprocessor, which is responsible for processing data used in the model training process. *Class methods*: loadDataFromDB - load data from the database; prepareData - prepare data before using it in the model training process.

The InferenceDataPreprocessor class. Implementation of the abstract class MLModelDataPreprocessor, which is responsible for processing data used in the process of making predictions of the initial data by the machine learning model. *Class methods*: loadDataFromDB - load data from the database; prepareData - prepare data before using them in the process of predicting the output data by the model.

The ApplicationConstants class. Saving constant values used in the application. *Class attributes*: OLX_SOURCE - a link to the OLX resource API for obtaining real estate data; OLX_API_KEY - a key to use the OLX resource; DIM_RIA_SOURCE - a link to the DimRia resource API for obtaining real estate data; DIM_RIA_API_KEY - key to use the DimRia resource; ADDRES_TO_COORDINATES_SOURCE - a link to a third-party API that converts the address to coordinates; GEO_DATA_SOURCE - a link to a third-party API that returns a list of locations near the specified geographic coordinates.

Here are the diagrams that explain how the process of buying and selling real estate works for users of this system. Activity diagrams have been constructed that describe step by step the process of buying (Figure 4) and selling (Figure 5) real estate when using the system.

The most important operation in the system under development is the valuation of real estate according to its parameters. Therefore, it is also advisable to consider the list of steps that need to be taken to implement this process. Activity diagram of the subprocess describing the real estate valuation operation is shown in Figure 6.

5. Developing a machine learning algorithm to address the task of predicting real estate values

The aim of the information system is to streamline the real estate search, valuation, and facilitate advantageous transactions between buyers and sellers. The primary objective of the project is to implement machine learning algorithms to execute the valuation function, catering to all participants in the real estate market. Simultaneously, attaining the utmost accuracy in forecasting is crucial. The target is to reach an average accuracy of 5-7% in determining real estate values. The error rate will be computed using test data. Achieving this level of accuracy will be deemed satisfactory, as the average cost of realtor services typically ranges from 3-5% of the total real estate selling price, thereby almost entirely compensating for any errors in the calculations.

5.1. Stages of developing a machine learning model and rationalizing approaches to address the issue

The creation of a machine learning model involves several stages, with the successful completion of each preceding stage greatly influencing the subsequent ones, as well as the overall final outcome.

The first stage:

- searching and downloading data from all available sources. For this purpose, the official API of the source is used;

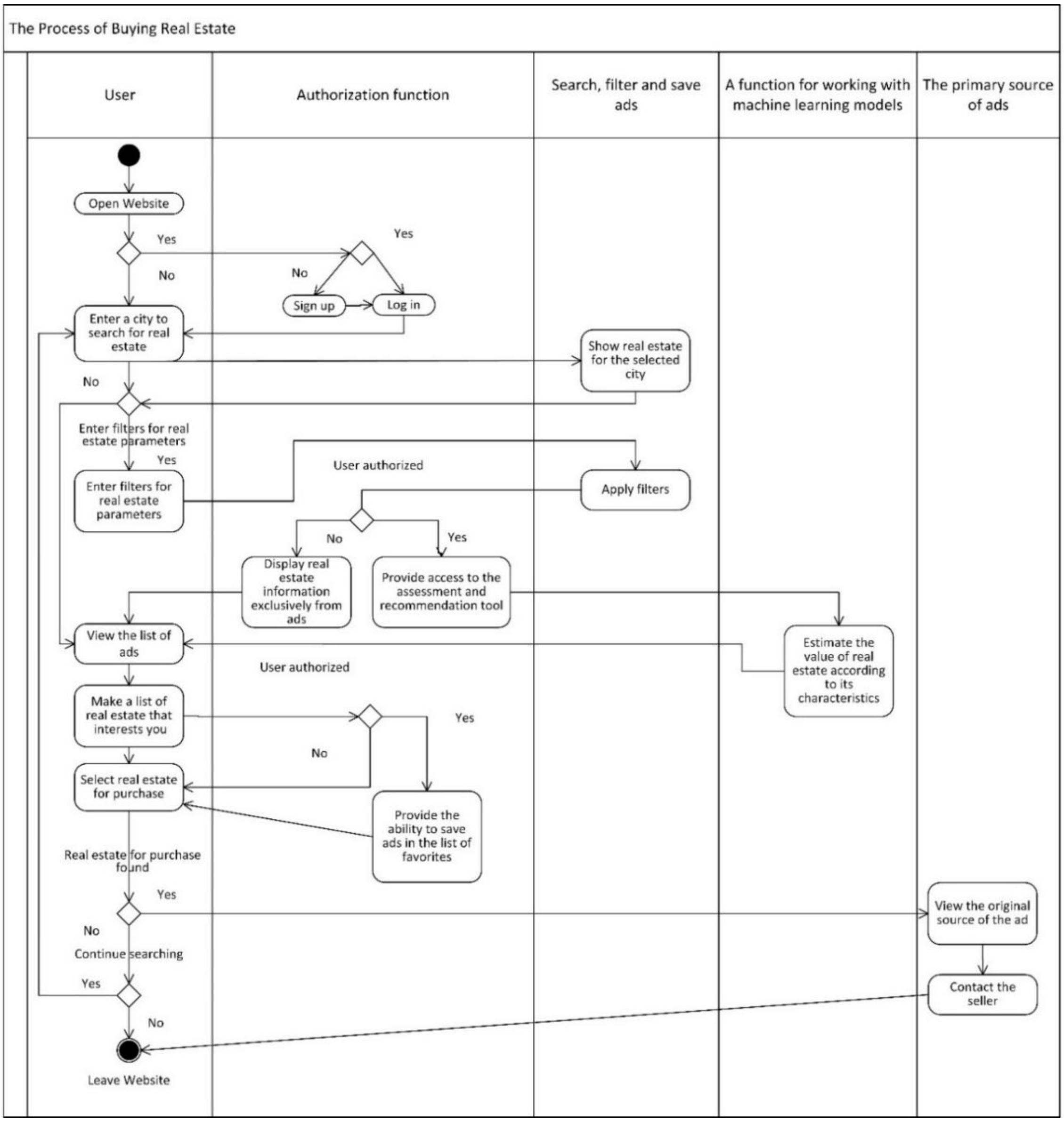

Figure 4: Activity diagram describing the process of buying real estate

- filtering data: finding and removing duplicate ads, as well as identifying outliers - data objects that stand out too much from the rest;
- checking if there is any missing data in the ads and what kind of data. If missing data is deemed relatively insignificant, it can be filled in; however, if it's deemed crucial, such records should be filtered out. Filling in missing values can be done using the mean (when the sample has a normal data distribution, no outliers or abnormal values), mode (when the distribution is not normal and the sample contains outliers), and median (to determine the category) of the total sample. The choice of the optimal approach to averaging values depends on the type of data and its distribution in the sample;

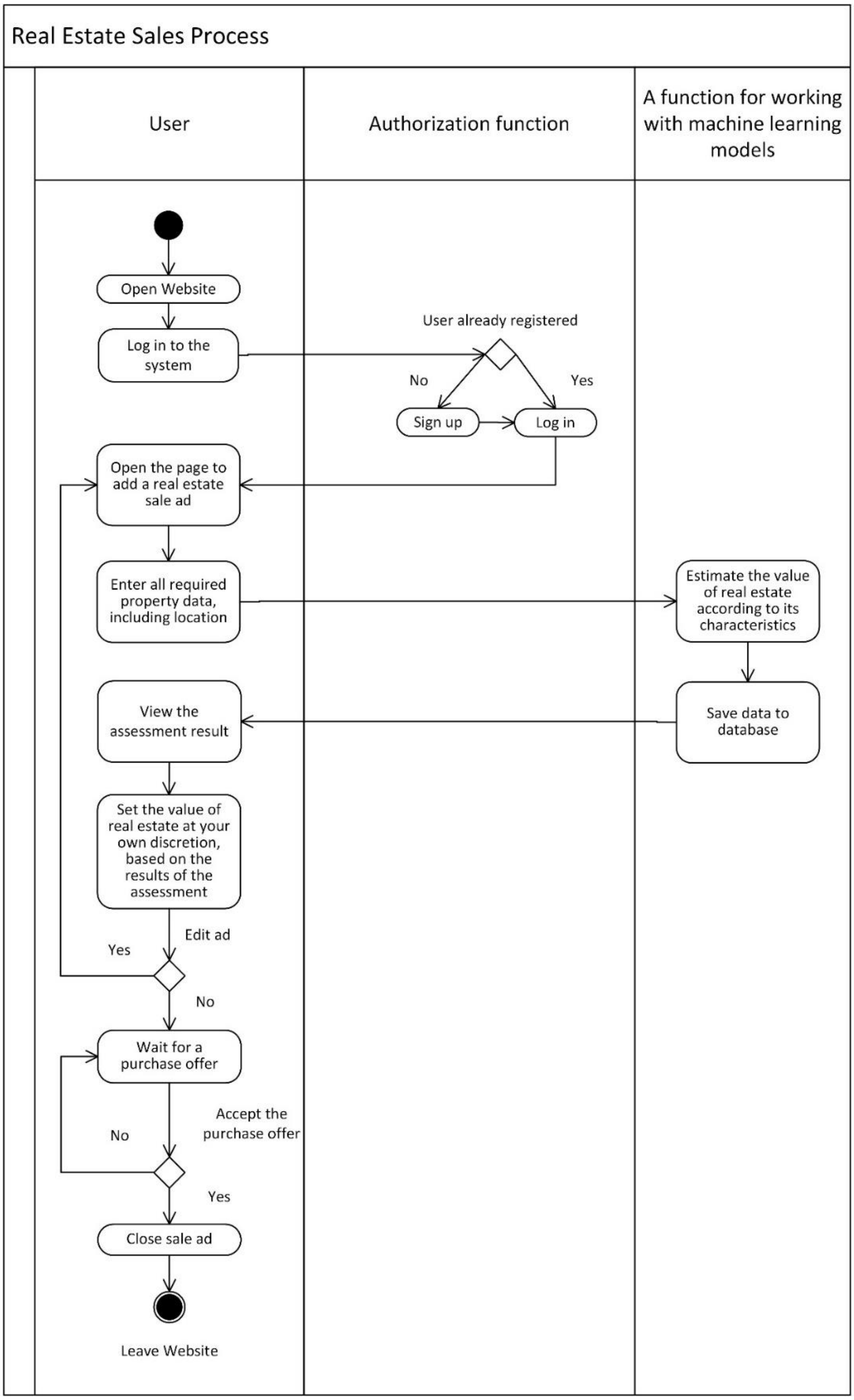

Figure 5: Activity diagram describing the process of selling real estate

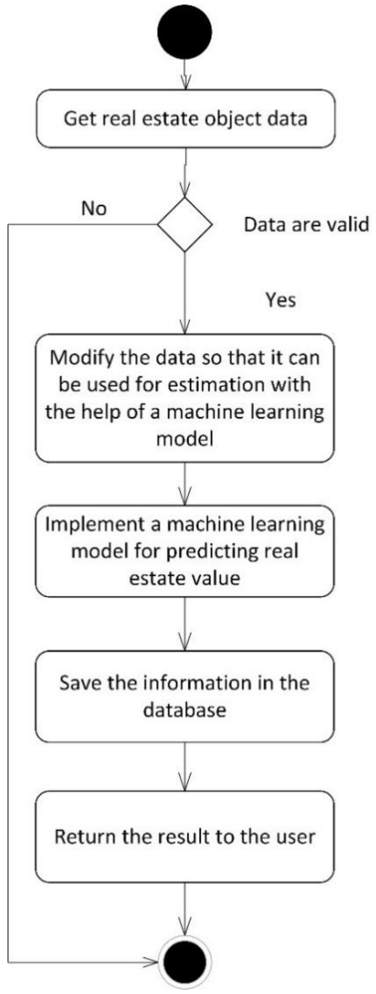

Figure 6: Activity diagram of the subprocess describing the real estate valuation operation

- data unification: converting them to the form used in the system;
- determining the coordinates of real estate by address using an external API;
- supplement the geographical data of the property: The count of nearby bus stops, hospitals, schools, and other amenities.

The second stage:

- identifying the minimal set of crucial features and parameters of real estate that significantly impact a machine learning model's ability to forecast real estate value. This process is conducted empirically, involving the modification of certain individual real estate characteristics to enhance their processing efficiency within the model. To achieve this, the One-Hot Encoding method is employed, which simplifies complex categorical features by representing them as binary values of 0 or 1.

The third stage:

- dividing data into two sets - training and testing, typically in an 80%-20% ratio. The testing data is chosen using cross-validation to ensure robust evaluation of the model's performance. The approach involves sequentially extracting 20% of the data as test samples from the entire dataset. The model is then trained using the same algorithm on the remaining data, with different subsets allocated for testing: 0-20%, 20-40%, 40-60%, 60-80%, and 80-100%. This iterative process ensures optimal training data selection and mitigates the risk of overfitting the model;
- training of machine learning models for a large city typically involves dealing with thousands to tens of thousands of active ads (prior to filtering). Given this relatively small dataset size, the expected training time for the model should ideally range from seconds to minutes.

The primary objective is to ascertain the value of real estate using specific input parameters, a task for which regression algorithms are instrumental [15-28]. Regression involves examining the connection between independent variables or features and a dependent variable or outcome. In the context of machine learning, regression serves as a method for predictive modeling, enabling the prediction or anticipation of results. In our analysis, we explored a variety of algorithms commonly used for regression tasks. These included linear regression [15-19], multiple linear regression (MLR) [20], decision trees [19, 23, 24], random forests [19, 29, 30], support vector regression (SVR) [19], and K-nearest neighbours (KNN) [31]. Regression techniques vary in their data handling capabilities. Some can handle a high number of independent variables, while others are better suited for specific data types. Machine learning regression models make varying assumptions about the relationships between independent and dependent variables. To achieve optimal results, avoid relying solely on a single algorithm. Instead, evaluate the performance of multiple approaches to identify the most efficient option. To efficiently compare different algorithms, we'll start with their default settings.

The fourth stage:

- evaluating model performance with an appropriate quality metric.

To analyze the impact of model complexity on accuracy, we will employ a regression quality metric [32]. To ensure consistency, a single metric was selected to evaluate model quality. After examining various popular metrics, R-squared was deemed the most suitable choice for this task.

R-squared, also known as the coefficient of determination, quantifies the proportion of variation in the dependent variable that can be attributed to the regression model. A key feature of this approach is its ability to quantify the benefit of using the full model with explanatory variables. This is achieved by comparing it to a simpler model that only predicts a constant value (no influence of input variables) and the input variables are missing or their regression coefficients are equal to zero.

A widely applicable formula exists for computing the coefficient of determination [34]:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y_i} - y_i)^2}{\sum_{i=1}^{n}(\overline{y_i} - y_i)^2}\ , \qquad (1)$$

where y_i – actual value; $\hat{y_i}$ – calculated by the model value; $\overline{y_i}$ – average value:

$$\overline{y_i} = \frac{1}{n}\sum_{i=1}^{n} y_i\ . \qquad (2)$$

R-squared (coefficient of determination) tells us how much improvement our regression model brings over a simple model that just predicts the average value for all data points.

In practice, the interpretation of the R-squared value is guided by the following scale [34]:

- R-squared values below 0.5 suggest a weak to moderate fit between the predicted and actual values;
- R-squared values greater than 0.5 indicate a moderate to strong fit between the predicted and actual values;
- An R-squared value exceeding 0.8 generally indicates a strong fit between the predicted and actual values, suggesting the model explains a large proportion of the variance.

It's evident from the formula that when the predicted values of $\hat{y_i}$ equal the arithmetic mean $\overline{y_i}$, the calculation in the fraction yields 1. Consequently, the coefficient of determination equals 0, indicating the model's poor performance. Conversely, if the predicted values $\hat{y_i}$ equal the actual values y_i, the numerator will be 0, resulting in a coefficient of determination of 1. This signifies that the model accurately predicts all values [34].

Once we've identified the most effective algorithms for a particular task, we proceed by selecting one or more of them and fine-tuning their hyperparameters. This enables us to attain a high level of accuracy tailored to the specific task at hand.

Certainly, in this approach, machine learning regression algorithms will be sourced from pre-existing libraries, given that the outlined sequence of steps entails concurrent utilization of multiple algorithms.

It was determined to gather all essential data solely for a single designated city due to the resource-intensive nature of the data collection process. Moreover, individual machine learning models must be trained for each city to ensure the program's accuracy. This necessity arises from the significant variation in real estate prices across different cities in Ukraine.

5.2. Selection and justification of tools for developing system architecture components

After researching the problem and algorithms for solving it in detail, evaluating existing competitive systems, determining the procedure for training a machine learning model, building diagrams and charts that describe the application's operation, and establishing requirements, you can finally proceed to the project development stage.

To facilitate the implementation of this system, a microservice architecture was chosen for its ability to provide simplicity, flexibility, and scalability [35-38]. The software product will comprise three distinct, interconnected services: the "Data providing service" the "Backend service" and the "Frontend service".

Data providing service (Figure 7). This service will be dedicated solely to data operations, including searching, filtering, and initial processing. Additionally, it will handle model training, testing, and real estate value prediction based on input parameters. Its responsibility also extends to regular data updates at specified intervals to monitor market dynamics and furnish the application with current data, crucial for advertisements.

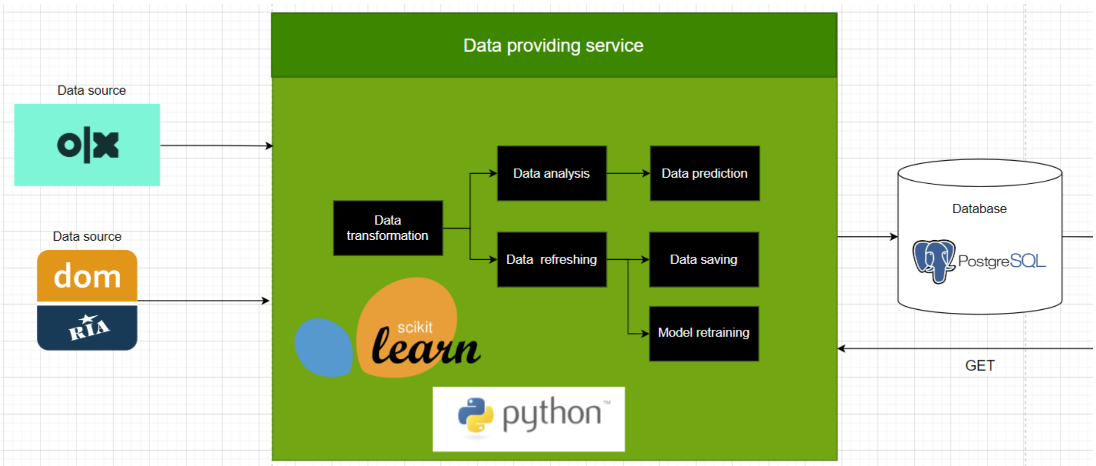

Figure 7: Scheme of operation of the "Data providing service" service

The data is stored in the database and is always available for other parts of the application.

Python is the best programming language for this task, as it is quite flexible and has a large number of mathematical libraries related to data processing and machine learning. The following libraries are used:

- Pandas – a library used for data processing and analysis. It provides access to data structures such as Series and DataFrame, as well as tools for working with them. These data structures are built on top of the NumPy library, which is another important library for numeric and array operations in Python.
- DataFrame – a two-dimensional labeled data structure that resembles a table. This structure consists of rows and columns, where each column can have a different type of data (for example, integers, floating point numbers, text). DataFrames can be considered as a container for Series objects.
- Series – a one-dimensional labeled array that can contain data of any type. You can work with it as a regular array, accessing data by index, as well as with an associated array, accessing data by key.

- Pyspark ML – a library that serves as an interface for working with Apache Spark. It provides access to machine learning algorithms, feature engineering tools, and machine learning model evaluators designed to work flawlessly with distributed data processing. It also includes a wide range of machine learning algorithms, including regression, classification, clustering, and collaborative filtering.
- Scikit-learn – one of the most popular libraries used in programs that involve working with machine learning. It contains tools for data processing, dimensionality reduction, anomaly detection, and provides a wide range of machine learning algorithms for classification and regression.
- OpenStreetMap Nominatim API – a no-cost service enabling retrieval of geographical coordinates for a location on the map using its complete address, as well as reverse geocoding, which provides the address information for a given set of coordinate.
- Python Overpass API – a free service that allows you to interact with the Overpass API service based on the OpenStreetMap project. It allows you to get information about the location of various geographical objects, such as public transport stops, parking lots, restaurants, buildings, etc. Using this service, you can also build navigation routes, determine distances between locations, and much more.

Backend service (Figure 8). The backbone of the application's business logic, this component facilitates the transmission of data to the "Data providing service" using parameters obtained from the user to ascertain property values. Additionally, it offers diverse functionalities for data analysis, processing, real estate recommendations, email notifications, data caching, and user authorization. Separating the logic in this way will give the system flexibility, allow it to scale more efficiently, add new and maintain existing functionality, and ensure simplicity. The development of this component leverages the Java programming language and the Spring Boot framework.

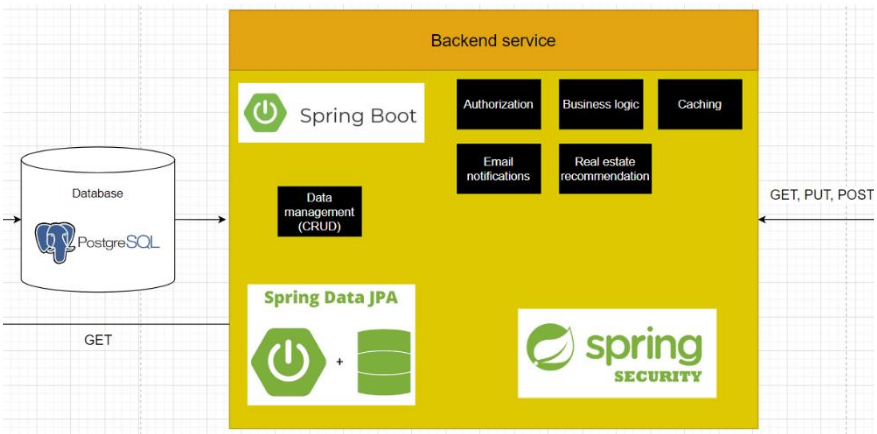

Figure 8: The scheme of the "Backend service" service

This approach is used because it is popular for creating web servers and applications used in microservice architecture. Its advantages are: quick creation and configuration of applications; simple database management; use of dependency injection and control inversion; a wide range of tools; advanced support and ease of testing; support for aspect-oriented programming; a large community, which means easy resolution of potential difficulties and problems [39].

Frontend service (Figure 9). This service will oversee the interaction between the application and end-users. Its primary role is to visualize data using various methods, present information in a user-friendly manner, and ensure intuitive navigation and usage of the application. Also, this service should correctly receive and transmit data for communication with other services. It will be implemented as a web application, as it allows for easy support and distribution of the project, provides accessibility from any user device.

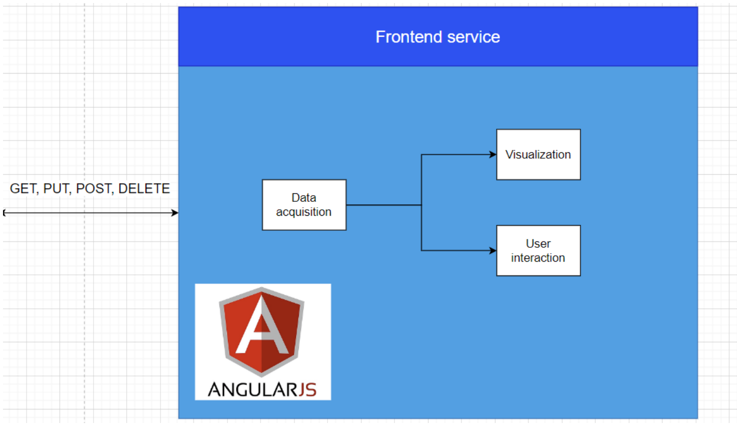

Figure 9: The "Frontend service" service working scheme

For the development of this component, HTML and CSS technologies were utilized alongside the JavaScript programming language, supplemented by one of its leading frameworks, Angular. This framework was chosen because of a number of its advantages that will be useful in the development of this system [40]:

- The declarative approach to application development provides a clear and organized way of representing the program structure and linking data, which can significantly increase the speed of development.
- The use of modules as units of system components, which makes it possible to easily maintain the code and reuse it. This is especially useful when the application grows to a large size.
- Integration with TypeScript, which improves code quality, reduces errors, and provides access to additional features. This makes it easier to maintain and refactor the code.
- Angular's two-way data binding is a powerful feature that makes it easy to synchronize data between the model and the view. It automatically updates the

user interface based on model changes and vice versa, which can save developers a significant amount of time and effort.

- Extended testing capabilities due to the modularity of the application, as well as the availability of special utilities. Provides quick creation of end-to-end, unit tests, and integration tests.

6. Testing of the newly developed algorithm for forecasting real estate values

6.1. Exploring, filtering and handling data

The initial and pivotal stage, setting the groundwork for all subsequent functionalities of the system, involves acquiring pertinent real estate market data. The volume and quality of data accessible for training and testing a machine learning model greatly influence its efficacy. Data can be of different types and come from different sources, such as databases, spreadsheets, or APIs [41].

It's essential to explore numerous sources of information to ensure an ample supply of data for constructing a high-caliber machine learning model. Simultaneously, regularly updating this data at specified intervals enhances the potential for in-depth analysis, facilitates notification functionalities, and enables continual model retraining with current data.

It was decided to collect all the necessary data only for one specific city (Kyiv), as the data collection process is very resource-intensive, and especially because a separate machine learning model needs to be trained for each city to ensure the program's accuracy [42]. Consequently, developing a single universal model for the entirety of Ukraine would likely fail to accurately estimate the cost of input data, resulting in unpredictable outcomes, particularly for outliers. To address this, various approaches to data collection were employed. The initial approach involved data mining through the official APIs of the DIM.RIA [7] and OLX [8] websites. Additionally, Telegram channels such as "Real Estate Kyiv and Region" [43] and "Real Estate of Kyiv Region" [44], which publish ads for property sales and leases, were utilized. Using the Telethon library [45], this data was collected and processed for subsequent utilization.

Upon data collection and processing, filtration becomes imperative. In cases where a property listing lacked values for kitchen or living space, the median of these values within the entire dataset was employed to populate the missing data. The arithmetic mean was deemed inappropriate due to the skewed distribution of this data. Instead, for categorical data types such as heating type, ad type, and wall type, the mode of the sample data was utilized for filling in missing values.

The following data was determined to augment the real estate information in Kyiv: kitchen area - 13 square meters, living area - 30 square meters, wall type - red brick, year of construction - 2013, ad type - from an intermediary, heating type - centralized..

In order to translate the address of the property into coordinates on the map, we used the open source OpenStreetMap Nominatim API [46]. We identify all significant locations within a specified radius near the property by employing an open-source data tool, the

Python Overpass API [47]. Utilizing this API necessitates the creation of a suitable query containing the coordinates of a specific point and the desired search radius.

We obtained 127 locations within a 100-meter radius of the address. Clearly, this data will require filtration, as the list encompasses various items such as trees, trash cans, street lamps, mailboxes, and benches. Nonetheless, among these, there are crucial locations that are likely to influence the value of real estate.

6.2. A machine learning model training process

With the data collected, filtered, and enriched with supplementary parameters, the next step involves initiating model training. It has been determined to employ multiple machine learning regression algorithms simultaneously to ascertain the one most adept at tackling the given task.

The algorithms were executed with default parameters, and the results were obtained (Figure 10).

```
LinearRegression             0.35640526160283725
DecisionTreeRegressor        0.3287186211464418
KNeighborsRegressor          0.2439190832334115
SupportVectorRegression      0.31189351581228564
GradientBoostingRegressor    0.6345036822381225
RandomForestRegressor        0.6815511209834788
```

Figure 10: Comparison of model accuracy trained with different algorithms using the R-squared metric

Upon comparing the coefficients of determination of the obtained results, it became evident that the "random forest" algorithm outperformed the others. Consequently, further model development and hyperparameter tuning will be conducted for this algorithm.

The subsequent hyperparameters were chosen empirically, yielding enhanced outcomes in forecasting the predicted values via the random forest algorithm. Through hyperparameter tuning and employing the cross-validation method, the R-squared value was elevated to 0.81, signifying a commendable achievement.

To assess the machine learning model, a validation dataset of real estate data was established and tested to evaluate the model's performance in predicting values against actual data (Figure 11).

In Figure 11, the vertical axis depicts the real estate value per square meter in US dollars, while the horizontal axis represents the proportion of the number of advertisements. The red points indicate the predicted values by the model, whereas the blue ones represent the actual values.

The graph illustrates a phenomenon known as heteroskedasticity, wherein as the dependent variable increases, the predicted variable exhibits larger deviations from the actual values. In other words, as the value of real estate per square meter increases, the variability of the predicted values also increases. This issue stems from the limited amount of data utilized during model training, which is a common occurrence.

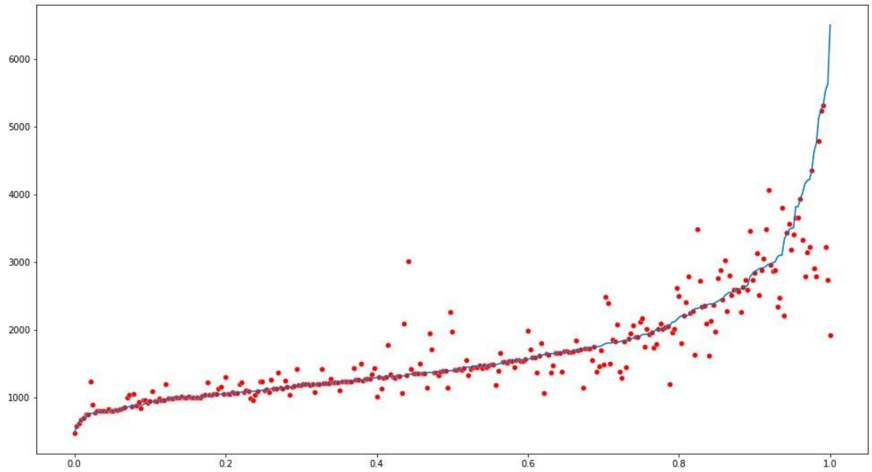

Figure 11: The outcome of forecasting the value of real estate per square meter in Kyiv based on its parameters using the developed model

The majority of properties in the dataset have a price per square meter of less than $2,000, making it challenging for the model to accurately predict such unique cases. In the future, addressing this problem can be achieved by gathering additional data on real estate for sale.

To enhance the assessment of the absolute error calculation results, the data are presented as follows:

APE > 50% for 0.012084592145015106 of test data
APE > 20% for 0.1933534743202417 of test data
APE > 10% for 0.31722054380664655 of test data
APE > 5% for 0.4108761329305136 of test data
APE > 1% for 0.5347432024169184 of test data
APE < 1% for 0.4652567975830816 of test data

Here, APE represents the absolute error determined by the formula:

$$APE = \left| \frac{d_i - \widehat{d_i}}{d_i} \right| , \quad (3)$$

where d_i – real value; $\widehat{d_i}$ – calculated by the model value.

Figure 12 illustrates the percentage of absolute error in forecasting real estate values. The average error stands at 8.49%, with a median of 1.9%.

In summary, regarding the machine learning model, it can be concluded that overall, the model performed admirably, as indicated by the study results. With the consistent accumulation of new data and subsequent retraining of the model, there is even the prospect of achieving improved performance.

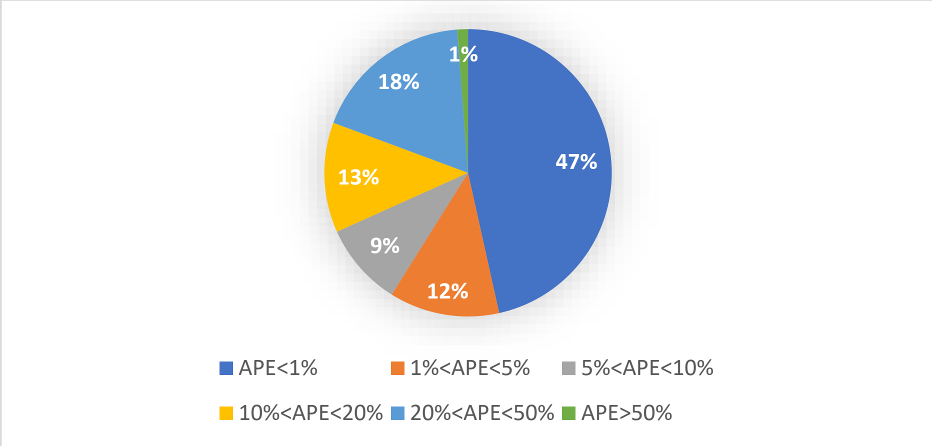

Figure 12: The absolute discrepancy in the forecasted real estate worth

By adhering to the same methodology, it is feasible to devise an effective model for any other city in Ukraine, provided there is a substantial volume of data available.

6.3. Evaluating and deploying microservices for application functionality

The subsequent stage involves testing the service serving as the system's server component.

To guarantee the utmost reliability of the service, unit tests were devised for all its classes and methods. These tests encompass various scenarios for both correct and incorrect requests, objects, parameters, and so forth. This not only ensures the system's predictability but also serves as a crucial factor for its subsequent maintenance and enhancement.

Additionally, we conducted manual testing of the service using the Swagger tool. We validated the accuracy of all program endpoints by executing various requests with different parameters. This testing process uncovered several minor errors, which were subsequently rectified.

Once the server component has been confirmed to function correctly, the next step is to commence testing the service responsible for the user aspect of the system. In this scenario, conducting end-to-end testing of all functionalities and capabilities of the service is most suitable. Upon successfully completing this testing stage, the system can be deemed tested and prepared for use.

Given that the system comprises three distinct microservices necessitating configured connections for information exchange and shared database usage, it's crucial to facilitate swift and straightforward installation, configuration, and execution of the system. Moreover, the services employ different technologies, such as Java, Node.js, and Python, which may require additional setup, environment variable creation, and similar adjustments.

The optimal solution to this challenge is to containerize each service separately using Docker tools. This method not only streamlines the deployment process but also enables scalability using orchestration tools. Similarly, the "Frontend service" and "Data providing service" were containerized, generating the respective Docker images for each.

Next, we create a docker-compose.yaml file, enabling the configuration of essential parameters and links between multiple containers.

With the latest versions of the images for all system services created and the corresponding configuration file in place, execute the following commands to initiate the system: "docker-compose build", and once completed, "docker-compose up".

Figure 13 depicts the interconnected containers of all individual services utilized in the system via Docker Compose. The servers launched successfully, affirming the accuracy of the configuration.

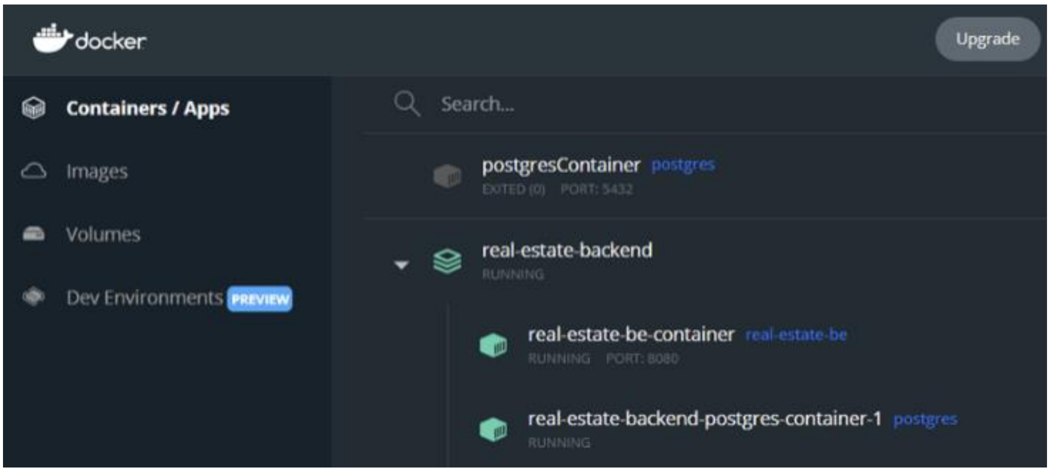

Figure 13: The outcome of system deployment via Docker Compose

Figures 14 and 15 depict the outcome of the successfully deployed system.

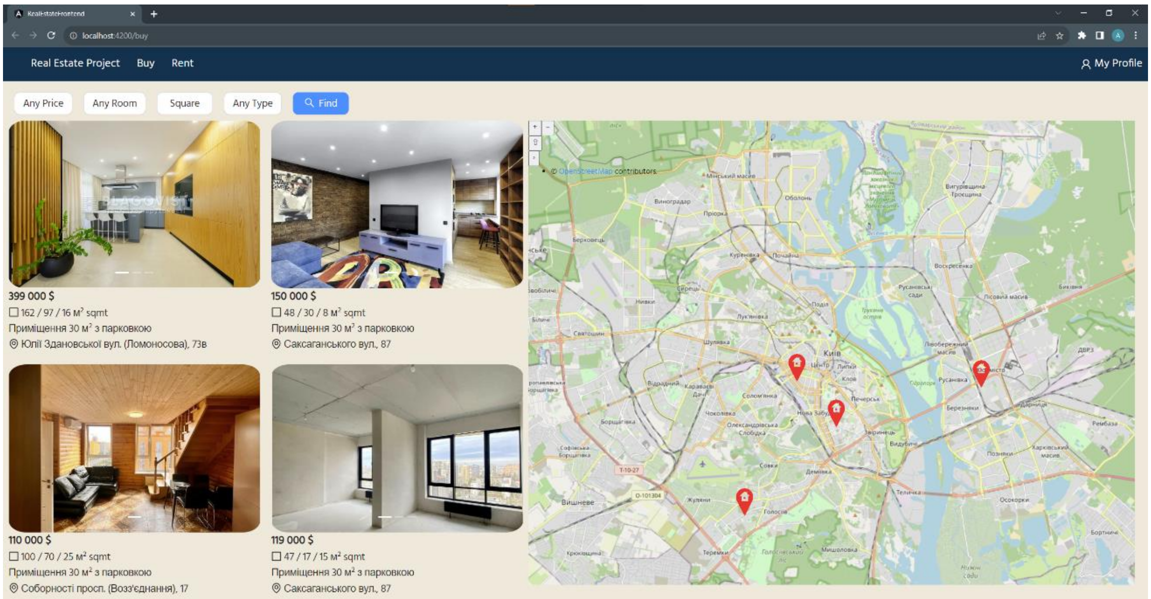

Figure 14: Real estate value estimate for the buyer

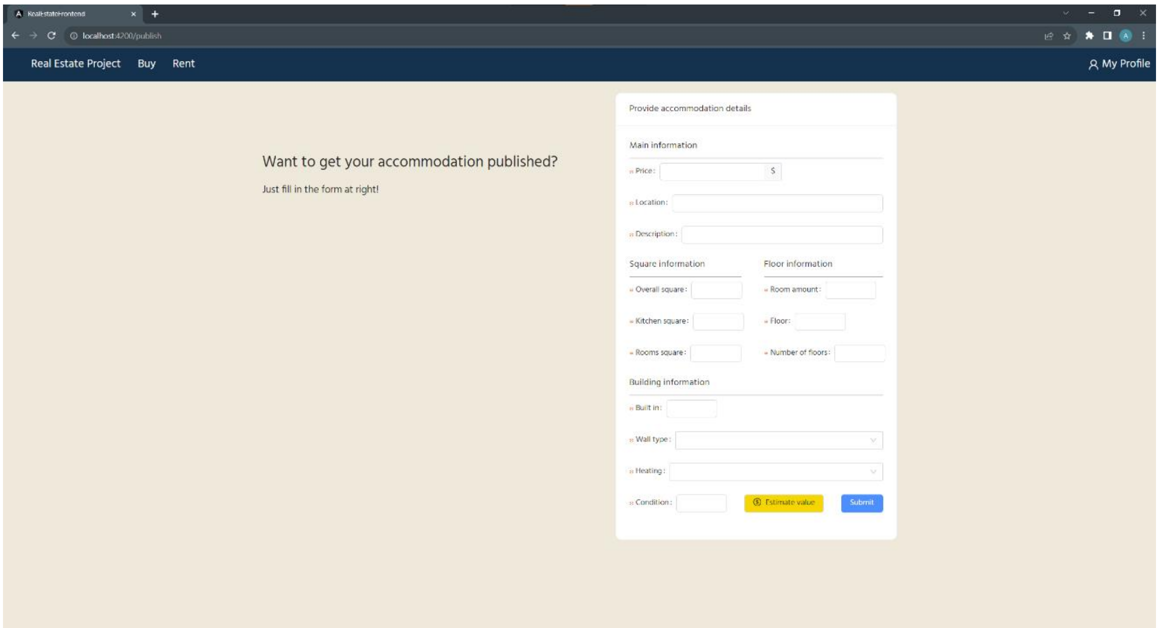

Figure 15: The outcome of real estate evaluation based on the parameters set in the system

It has been confirmed that all services operate correctly and interact seamlessly with one another. Data is accurately transmitted, stored, and presented. This marks the concluding stage of verification, confirming the system's readiness for utilization.

7. Conclusions

The study's outcome is a machine learning model capable of determining real estate value based on its physical parameters and geographical location. The primary advantages of employing artificial intelligence methods for predicting real estate value were analyzed.

We examined both Ukrainian and international applications addressing the challenge of value forecasting and real estate valuation.

A distinctive feature of our development is that the model training process incorporates not only basic real estate characteristics but also geospatial data. This includes considering the presence and quantity of specific types of locations within a specified radius of the property, thereby enhancing the accuracy of value predictions.

The process of constructing a machine learning model is conceptually divided into four stages: data collection, filtering, processing, supplementation, partitioning into various samples, and training the model based on these datasets. Regression methods, algorithms, and quality metrics pertinent to predicting real estate value are explored. The model's efficacy is assessed on a validation dataset, and the prediction results along with the absolute error are visualized using graphs and charts. Analysis of this data leads to the conclusion that the model's performance aligns with the system's specifications, thus signifying the success of the study.

The developed information system will streamline and expedite the process of discovering optimal real estate for purchase, as well as facilitating its assessment for future sale. Real estate sellers will promptly evaluate their property's worth based on parameters, location, and prevailing market conditions, while buyers will access a list of recommended properties priced in line with their actual attributes or below market value.

References

[1] The Importance of Accurate Property Valuation in Real Estate, 2023. URL: https://sugermint.com/the-importance-of-accurate-property-valuation-in-real-estate/.

[2] AI in real estate property valuation: Is it really a game-changer?, 2023 URL: https://mdevelopers.com/blog/ai-real-estate-property-valuation.

[3] I. Kolesnikova, Using Artificial Intelligence for Real Estate: A Comprehensive Guide, 2023. URL: https://mindtitan.com/resources/industry-use-cases/artificial-intelligence-in-real-estate/.

[4] REAL-TIME PROPERTY VALUATIONS: HOW AI ALGORITHMS ARE MAKING IT POSSIBLE, 2023. URL: https://www.realspace3d.com/blog/real-time-property-valuations-how-ai-algorithms-are-making-it-possible/.

[5] O. Veres, P. Ilchuk, O. Kots, Data Science Methods in Project Financing Involvement, International Scientific and Technical Conference on Computer Sciences and Information Technologies 2 (2021). 411–414. doi: 10.1109/CSIT52700.2021.9648679.

[6] O. Veres, P. Ilchuk and O. Kots, Data Analytics on Debt Financing Research Based on Scopus and WoS Metrics, In 2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT), Lviv, Ukraine, 2023. doi: 10.1109/CSIT61576.2023.10324179.

[7] DIM.RIA – all real estate of Ukraine. Sale and rent of any real estate, 2024. URL: https://dom.ria.com/uk/.

[8] OLX.ua ads: Ukrainian classifieds service, 2024. URL: https://www.olx.ua/uk/nedvizhimost/.

[9] Zillow.Com, Agents. Tours. Loans. Homes, 2024. URL: https://www.zillow.com/.

[10] Realtor.Com, Homes for Sale, Real Estate & Property Listing, 2024. URL: https://www.realtor.com/.

[11] Redfin.Com, Real Estate, Homes for Sale, MLS Listings, Agents, 2024. URL: https://www.redfin.com/.

[12] PropStream.Com, Most Trusted Provider of Real Estate Information, 2024. URL: https://www.propstream.com/.

[13] N. Berezhna, Buying a home: do you need a realtor and how much do his services cost in Ukraine, 2021. URL: https://realestate.24tv.ua/kupivlya-zhitla-potriben-rieltor-skilki-koshtuyut-ostanni-novini_n1525065.

[14] I. Hrabynskyi, I. Prykhodko, V. Halanets, L. Prokopyshyn-Rashkevych, A. Adamovsky and I. Zhygalo, The Impact of the Russian-Ukrainian War on the Development of the Primary Residential Real Estate Market in Ukraine: Results of a Cluster Analysis. "Economic Affairs (New Delhi)" 67(4) (2022) 837-849. doi:10.46852/0424-2513.4s.2022.17.

[15] D. Castillo, Machine Learning Regression Explained, 2021. URL: https://www.seldon.io/machine-learning-regression-explained.
[16] Gelman, Andrew, and Jennifer Hill. Data analysis using regression and multilevel/hierarchical models. Cambridge university press, 2006. Gelman, A.; Hill, J. Data Analysis Using Regression and Multilevel/Hierarchical Models; Cambridge University Press: New York, NY, USA, 2006.
[17] Finch, W. Holmes, Jocelyn E. Bolin, and Ken Kelley. Multilevel modeling using R. Crc Press, 2019.
[18] Evans, Clare R., George Leckie, and Juan Merlo. "Multilevel versus single-level regression for the analysis of multilevel information: the case of quantitative intersectional analysis." Social Science & Medicine 245 (2020) 112499.
[19] What is Simple Linear Regression in Machine Learning?, 2023. URL: https://www.simplilearn.com/what-is-simple-linear-regression-in-machine-learning-article.
[20] Maulud, Dastan, and Adnan M. Abdulazeez. "A review on linear regression comprehensive in machine learning." Journal of Applied Science and Technology Trends 1.4 (2020) 140-147.
[21] D. Polzer, 7 of the Most Used Regression Algorithms and How to Choose the Right One, 2021. URL: https://towardsdatascience.com/7-of-the-most-commonly-used-regression-algorithms-and-how-to-choose-the-right-one-fc3c8890f9e3.
[22] Montgomery, Douglas C., Elizabeth A. Peck, and G. Geoffrey Vining. Introduction to linear regression analysis. John Wiley & Sons, 2021.
[23] C. Dawson, Understanding Multiple Linear Regression, 2021. URL: https://medium.com/swlh/understanding-multiple-linear-regression-e0a93327e960.
[24] Mahaboob, B., et al., A study on multiple linear regression using matrix calculus, Advancecs in Mathematics Scientifc journal 9.7 (2020) 1-10.
[25] S. Bouzebda, Y. Souddi, F. Madani, Weak Convergence of the Conditional Set-Indexed Empirical Process for Missing at Random Functional Ergodic Data. Mathematics 12 (2024). doi: 10.3390/math12030448.
[26] Y. Zhou, D. He, Multi-Target Feature Selection with Adaptive Graph Learning and Target Correlations. Mathematics 12 (2024). doi:10.3390/math12030372.
[27] T. Li, K. A. Frank, M. Chen, A Conceptual Framework for Quantifying the Robustness of a Regression-Based Causal Inference in Observational Study. Mathematics 12 (2024). doi:10.3390/math12030388.
[28] Leyland, Alastair H., and Peter P. Groenewegen. Multilevel modelling for public health and health services research: health in context. Springer Nature, 2020.
[29] Measure of impurity, 2019. URL: https://medium.com/@viswatejaster/measure-of-impurity-62bda86d8760.
[30] Bentéjac, Candice, Anna Csörgő, and Gonzalo Martínez-Muñoz. "A comparative analysis of gradient boosting algorithms." Artificial Intelligence Review 54 (2021) 1937-1967.
[31] P. Rishit, Understanding K-Nearest Neighbors: A Simple Approach to Classification and Regression, 2023. URL: https://pub.towardsai.net/understanding-k-nearest-neighbors-a-simple-approach-to-classification-and-regression-e4b30b37f151.

[32] Chicco, Davide, Matthijs J. Warrens, and Giuseppe Jurman. "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation." PeerJ Computer Science 7 (2021) e623.
[33] P. Bhandari, Central Tendency | Understanding the Mean, Median & Mode, 2023. URL: https://www.scribbr.com/statistics/central-tendency/ .
[34] What Is R Squared And Negative R Squared, 2018. URL: http://www.fairlynerdy.com/what-is-r-squared/ .
[35] L. Silva, Why Spring Boot is so popular: all about the framework, 2023. URL: https://www.linkedin.com/pulse/why-spring-boot-so-popular-all-framework-leonardo-holanda-e-silva.
[36] H. Dhaduk, Angular vs React: Which to Choose for Your Front End in 2023? 2023. URL: https://www.simform.com/blog/angular-vs-react/.
[37] O. Veres, N. Kunanets, V. Pasichnyk, N. Veretennikova, R., Korz, A. Leheza, Development and Operations - the Modern Paradigm of the Work of IT Project Teams. In 2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT) 3 (2019) 103-106. doi: 10.1109/STC-CSIT.2019.8929861.
[38] O. Veres, P. Ilchuk, O. Kots, Y. Levus, O. Vlasenko, Recommendation System for Leisure Time-Management in Quarantine Conditions, CEUR Workshop Proceedingsthis 3312 (2022) 263-282.
[39] Why Spring Boot is so popular: all about the framework, 2023. URL: https://www.linkedin.com/pulse/why-spring-boot-so-popular-all-framework-leonardo-holanda-e-silva.
[40] Angular vs React: Which to Choose for Your Front End in 2023?, 2023. URL: https://www.simform.com/blog/angular-vs-react/.
[41] ML | Introduction to Data in Machine Learning, 2023. URL: https://www.geeksforgeeks.org/ml-introduction-data-machine-learning/.
[42] What are the average prices on the secondary housing market in Ukraine: how much will you have to pay for a one-room apartment, 2023. URL: https://sud.ua/uk/news/ukraine/259841.
[43] Real estate Kyiv and region. 2024. URL: https://t.me/ppbestate.
[44] Real estate of the Kyiv region, 2024. URL: https://t.me/Neruhomist_Kyiv_region.
[45] Telethon's Documentation, 2024. URL: https://docs.telethon.dev/en/stable/
[46] Nominatim 4.3.0 Manual, 2024. URL: https://nominatim.org/release-docs/latest/api/Overview/.
[47] Python Overpass API, 2024. URL: https://python-overpy.readthedocs.io/en/latest/.