Predicting Apartment Renting Prices in Kosovë with Machine Learning Algorithms

A machine learning project by Uran Lajçi

A View from Prishtina with credits from 2023 Sailingstone Press LLS [16].

Tables

Figures

Abbreviations

LOR - Logistic Regression
LR - Linear Regression
DT - Decision Trees
RF - Random Forest
KNN - K-Nearest Neighbors
NB - Gaussian Naive Bayes
ACC - Accuracy
CV - Cross-Validation
MB - Mega Byte
HTML - Hypertext Markup Language
CSV - Comma Seperated Values
NaN - Not a Number

Libraries

The programming lanuguage used to train the models and build the application is Python [2]. The libraries used that need to be installed are:

- Streamlit [3]
- Joblib [4]
- Pandas [5]
- NumPy [6]
- Requests [7]
- Beautiful Soup [8]
- CSV [9]
- Time [10]
- Matplotlib [11]
- Seaborn [12]
- Scikit-learn [13]
- Threadpoolctl [14]

The complete code, files, models, and resulsts can be found in this GitHub repository [15].

Contents

Problem Definition

The problem at hand is the absence of apartment price predictions for apartments in Kosovë. The objective of this project is to develop an application that allows users to input apartment features and obtain the corresponding rental price. To achieve this, the project will utilize machine learning regression and classification algorithms to predict apartment prices based on the provided features. Additionally, the application will be built using the Streamlit framework to provide a user-friendly interface for easy interaction and price retrieval. By addressing this problem, the project aims to offer a valuable tool for both apartment seekers and landlords in Kosovë, facilitating informed decision-making and improving the efficiency of the rental market.

Data Acquisition

To solve the problem of predicting apartment rental prices in Kosovë, it was necessary to gather the required data since there were no existing datasets available for this specific domain. The following approach was employed to collect the data:

Data Source
The data was obtained by utilizing the BeautifulSoup Python library to scrape information from the website Gjirafa Patundshmeri [1].

Scraping Process
The BeautifulSoup library allowed for the extraction of relevant details from the website's listings. By navigating through the pages and parsing the HTML structure, the necessary data points were extracted.

Dataset Size
In total, over 34,000 data points were gathered during the scraping process. Each data point represents a distinct apartment listing from the website.

Attributes
The following attributes were collected for each apartment listing:

- "title": The title or headline of the apartment listing.
- "region": The region that represents the city of the apartment.
- "number_of_rooms": The number of rooms in the apartment.
- "quadrat": The size of the apartment in square meters.
- "price": The rental price of the apartment.
- "date": The date when the apartment listing was published.

By acquiring this dataset, comprising the aforementioned attributes, the project has obtained the necessary information to proceed with the development and evaluation of the machine learning regression and classification algorithms for predicting apartment rental prices in Kosovë.

id	title	region	number_of_rooms	quadrat	date	price
0	Banese me qira ne lagjen Bregu i Diellit	Prishtine	1	65m 2	11/07/2023	250 €
1	Banese me qira ne lagjen Arberia_2+1	Prishtine	2	90m 2	11/07/2023	500 €
2	Banese me qira ne lagjen Tophane	Prishtine	2	80m 2	11/07/2023	300 €
3	Banese me qira ne Lagje te Spitalit_2+1	Prishtine	2	82m 2	11/07/2023	330 €
4	Banese me qira ne Rrugen B_2+1	Prishtine	2	83m 2	11/07/2023	400 €

Table 1. The first five data points of the raw dataset.

Data Preprocessing

From the Table 1 we see that in order to use this dataset to make predictions we need to clean and preprocess some of the columns. Also from the raw data we can create some attributes that will be more useful for our purposes.

The title attribute is renamed to property_description besause it is a more descriptive name for the data that it contains. The data points that contain the value shitet in this column are removed, because they represent the apartements that are for sale and not for renting. Also the characters are lowercased and the non-alphanumeric characters are removed, and the leading and trailing whitespaces are removed. We are not going to use the title attribute in our prediction but these preprocesing steps are taken to make the values more readeble.

After exploring the region attribute values we see that it contains these cities of Kosova: 'Drenas', 'Ferizaj', 'Fushe Kosove', 'Gjakove', 'Gjilan', 'Kline', 'Lipjan', 'Malisheve', 'Mitrovice', 'Peje', 'Prishtine', 'Prizren', 'Vushtrri'. The other values where anomalies and un-precise values and where removed. But the data points that are apartemnents for the region of Prishtina are much more than every city, so we named all the regions other than Prishtina as other.

In the number_of_rooms attribute we removed all the non-numeric values, and we removed all the data points that have negative values for the number_of_rooms or that have a bigger number that 6. We removed the values less than 0 or bigger than 6 because after the exploration of the number_of_rooms showed that almost absolute number of rooms are from 0 to 6. Also we renamed this column to number of rooms to make it more readeble.

The 'quadrat' attribute initially combined numeric data and "m 2" unit. The first preprocessing step removed 'm 2', leaving numeric strings, which were then converted to numbers using pandas' 'to_numeric' function, replacing unconvertible values with NaN. Some square meter data was missing or unrealistically low (<=1); these were handled by assigning the mean square meter value of the corresponding room category. Finally, the attribute was renamed 'quadrat (m^2)' to indicate its numeric nature and the measurement unit.

The raw price data in this attribute was a combination of the price amount and the currency symbol "€". To clean this, the '€' symbol was first removed using a replace operation, similar to what was done with 'quadrat', leaving behind only the numeric part. After this, the data

was again converted into a numeric type using the same method as before. The price values were then filtered to only include apartments with a price range from 60 to 2000 euros. This filtering might have been done based on some business understanding or based on the exploration of the price data which indicated that most prices fell within this range. Lastly, the column was renamed from 'price' to 'price (euro)' to clearly indicate that the values represent the price of the apartments in euros.
The date attribute whas cleaned from the non-date values and its tybe was converted to datetime.

All that we did until now was preprocess the existing data and attributes. But these attributes are not enough for our prediction, and we see that we can engineer better attributes from the existing.

So, from the region attribute we did one-hot encoding and got 2 attributes region_Prishtine and region_Other region in kosove, this makes the region data more useful for prediction.

From the date attibute we created a attribute named seasons, that represents the season of the year: spring, summer, autumn, winter. This is done because from the domain knowledge shows that the based on the time of the year the renting price changes. After we got the season, we created four attributes season_spring, season_summer, season_autumn, season_winter with one-hot encoding to make the data more acceseble in the training and testing of the machine learning model.

id	property_description	number of rooms	quadrat (m^2)	date	price (euro)	region_Prishtine
0	banese me qira ne lagjen bregu i diellit	1	65	7/11/2023	250	1
1	banese me qira ne lagjen arberia21	2	90	7/11/2023	500	1
2	banese me qira ne lagjen tophane	2	80	7/11/2023	300	1
3	banese me qira ne lagje te spitalit21	2	82	7/11/2023	330	1
4	banese me qira ne rrugen b21	2	83	7/11/2023	400	1

Table 2. The first seven attributes of the first five data points after the preprocessing steps.

region_other region in kosove	seasons_Autumn	seasons_Spring	seasons_Summer	seasons_Winter
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0
0	0	0	1	0

Table 2. The last five attributes of the first five data points after the preprocessing steps.

In the Table 2 and Table 3 are shown the data after the preprocessing steps tha we explained in this section.

Data Exploration

Basic information about the dataset:
RangeIndex: 31933 entries, 0 to 31932
Data columns (total 12 columns):

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	31933 non-null	int64
1	property_description	31933 non-null	object
2	number of rooms	31933 non-null	int64
3	quadrat (m^2)	31933 non-null	float64
4	date	31933 non-null	object
5	price (euro)	31933 non-null	float64
6	region_Prishtine	31933 non-null	int64
7	region_other region in kosove	31933 non-null	int64
8	seasons_Autumn	31933 non-null	int64
9	seasons_Spring	31933 non-null	int64
10	seasons_Summer	31933 non-null	int64
11	seasons_Winter	31933 non-null	int64

dtypes: float64(2), int64(8), object(2)
memory usage: 2.9+ MB

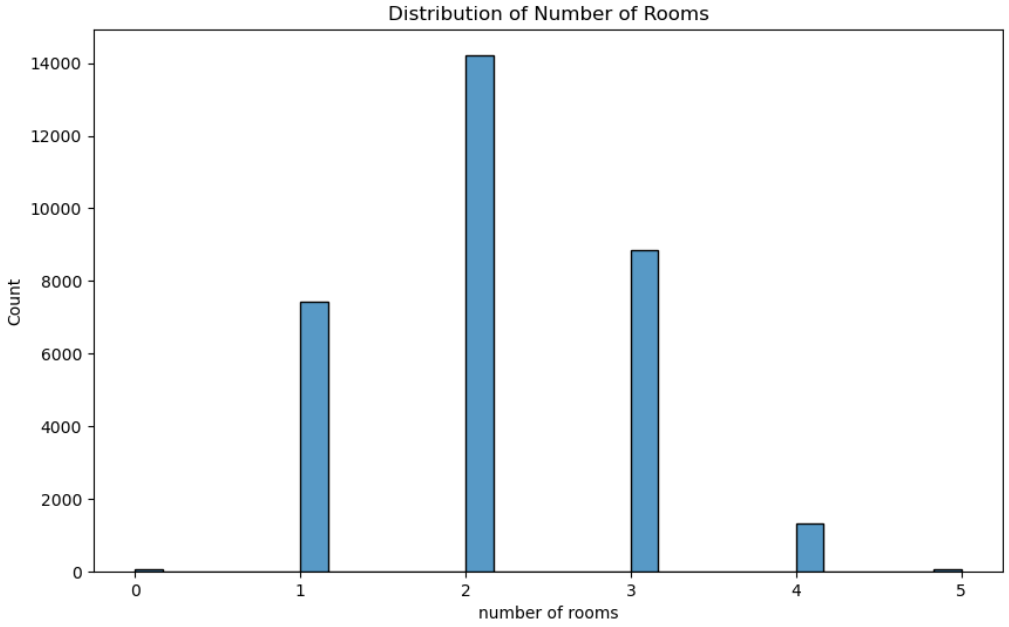

Figure 1. The distribution of the number of rooms in the apartmnents.

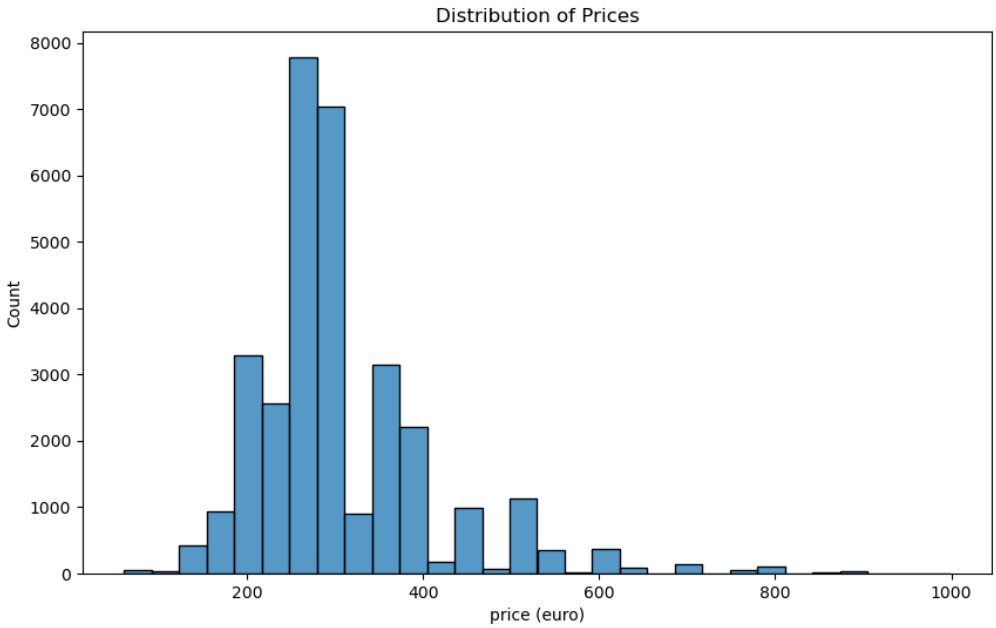

Figure 2. The distribution of the apartemnent renting prices.

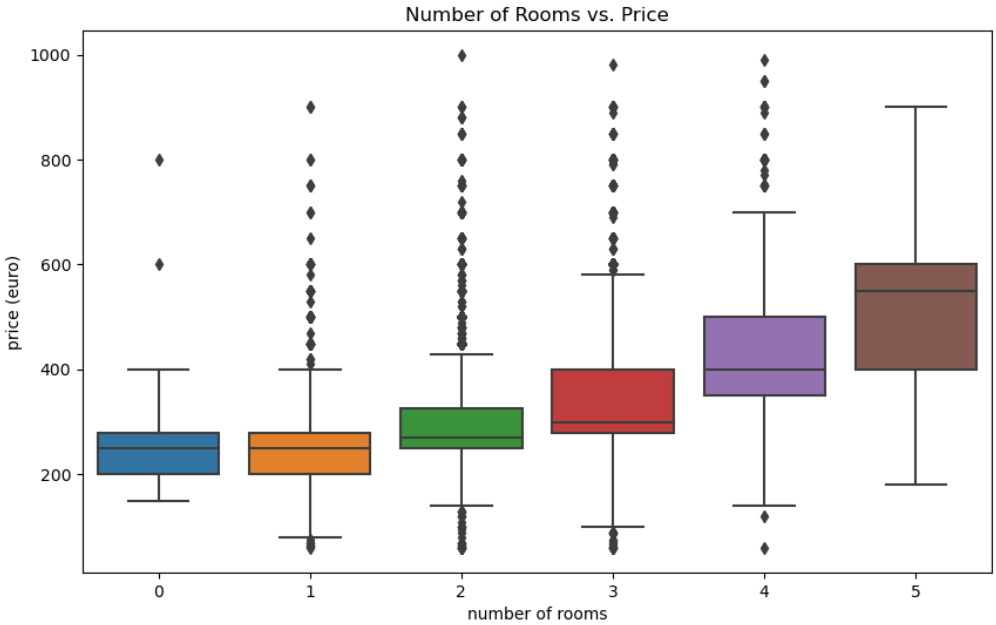

Figure 3. The relationship between the number of rooms and prices.

Predicting Renting Prices with Machine Learning Regression Algorithms

The renting_price_prediction_with_regression.py file uses machine learning algorithms to predict apartment renting prices in Kosovo. This solution utilizes five different predictive algorithms, including:

1. Linear Regression
2. Decision Trees
3. Random Forest
4. K-Nearest Neighbors (KNN)
5. Gaussian Naive Bayes

These algorithms are applied and evaluated on a preprocessed dataset, specifically designed for this task. Below is a breakdown of how the code works for the example of the Linear Regression algorithm:

Import Libraries
The necessary Python libraries are imported at the beginning of the script. These libraries provide functionalities for data manipulation (pandas, numpy), machine learning (sklearn), and performance optimization (threadpoolctl, warnings).

```
import pandas as pd
import numpy as np
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
from math import sqrt
from joblib import dump
import threadpoolctl
import warnings
```

Preprocessing
The code sets a limit on the number of threads that can be used by OpenBLAS, in order to prevent CPU over-utilization. Also, warnings are ignored for readability purposes.

```
warnings.filterwarnings("ignore")
threadpoolctl.threadpool_limits(limits=1)
```

Data Loading
The preprocessed dataset is loaded into a pandas DataFrame.

```
df = pd.read_csv("datasets/preprocessed_apartment_renting_data.csv")
```

Feature Selection
The feature matrix `X` and the target vector `y` are defined. The features include information like the number of rooms, square meters, region, and season.

```
X = df[['number of rooms', 'quadrat (m^2)', 'region_Prishtine',
'region_other region in kosove', 'seasons_Autumn', 'seasons_Spring',
'seasons_Summer', 'seasons_Winter']]
```

```
y = df['price (euro)']
```

Model Training
A Linear Regression model is created and trained using the dataset.

```
linear_regression_model = LinearRegression()
linear_regression_model.fit(X, y)
```

Model Saving
The trained Linear Regression model is saved to a file for future use.

```
dump(linear_regression_model, 'regression models/
linear_regression_model.joblib')
```

Predictions & Evaluation
The trained model is used to predict apartment renting prices, and the predictions are evaluated using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

```
linear_regression_predictions = linear_regression_model.predict(X)
linear_regression_mae = mean_absolute_error(y,
linear_regression_predictions)
linear_regression_mse = mean_squared_error(y,
linear_regression_predictions)
linear_regression_rmse = sqrt(linear_regression_mse)
```

Cross-Validation
A 5-fold cross-validation is performed to assess the model's performance and the average RMSE is calculated for these cross-validation folds.

```
linear_regression_scores = cross_val_score(linear_regression_model, X,
y, cv=5, scoring='neg_mean_squared_error')
linear_regression_avg_cross_val_score =
np.mean(np.sqrt(np.abs(linear_regression_scores)))
```

This process is repeated for each of the machine learning algorithms used in this project, namely Decision Trees, Random Forest, K-Nearest Neighbors, and Gaussian Naive Bayes.

Metric	**LR**	**DT**	**RF**	**KNN**	**NB**
MAE	65.39	57.29	56.72	61.96	80.26
MSE	8456.71	7119.15	6783.96	7907.44	13054.33
RMSE	91.96	84.37	82.36	88.92	114.25
CV	95.03	87.33	85.24	95.85	111.09

Table 4. The results of the regression models with 70-30 split.

Predicting Renting Prices with Machine Learning Classification Algorithms

In the renting_price_prediction_with_classification.py file, different machine learning models are employed to categorize apartment renting prices into predefined groups. The classification models used include:

1. Logistic Regression
2. K-Nearest Neighbors (KNN)
3. Decision Trees
4. Gaussian Naive Bayes
5. Random Forest

Data Preparation
Prices are categorized into six groups: '60-120', '120-180', '180-240', '240-300', '300-360', and '360+' in order to make the problem suitable for classification. These groups are created with the `pd.cut()` function.

The features used for the prediction ('number of rooms', 'quadrat (m^2)', regions, and seasons) are extracted into the variable `X`, and the newly created 'price_group' column becomes the target variable `y`.

Model Training, Evaluation and Saving
Each of the five models are trained, evaluated and saved in a similar manner. As an illustration, let's go through the process with the Logistic Regression model:

1. Model Training: The Logistic Regression model is trained using the `fit()` method on `X` and `y`.
2. Model Saving: The trained model is saved into a file using the `joblib.dump()` function for future use.
3. Evaluation: The model is evaluated on the same dataset it was trained on using accuracy as the metric. Predictions are made on `X` using the `predict()` method. The accuracy of the predictions is then calculated using the `accuracy_score()` function from `sklearn.metrics`.
4. Cross-Validation: The model's performance is further evaluated using 5-fold cross-validation. The `cross_val_score()` function from `sklearn.model_selection` is used, with the scoring parameter set to 'accuracy'. The average of these scores is then computed to give a more robust estimate of the model's ability to generalize to unseen data.

These steps are repeated for each model (KNN, Decision Trees, Gaussian Naive Bayes, and Random Forest). By comparing the accuracy and cross-validation scores, we can assess which model performs best on the provided dataset.

Metric	**LOR**	**DT**	**RF**	**KNN**	**NB**
ACC	0.46	0.53	0.54	0.45	0.20
CV	0.46	0.50	0.50	0.42	0.19

Table 5. The results of the classification models with 70-30 split.

Guide to the use of the Streamlit Application

To execute the steamlit application open the application.py file and run it with this command: streamlit run application.py.

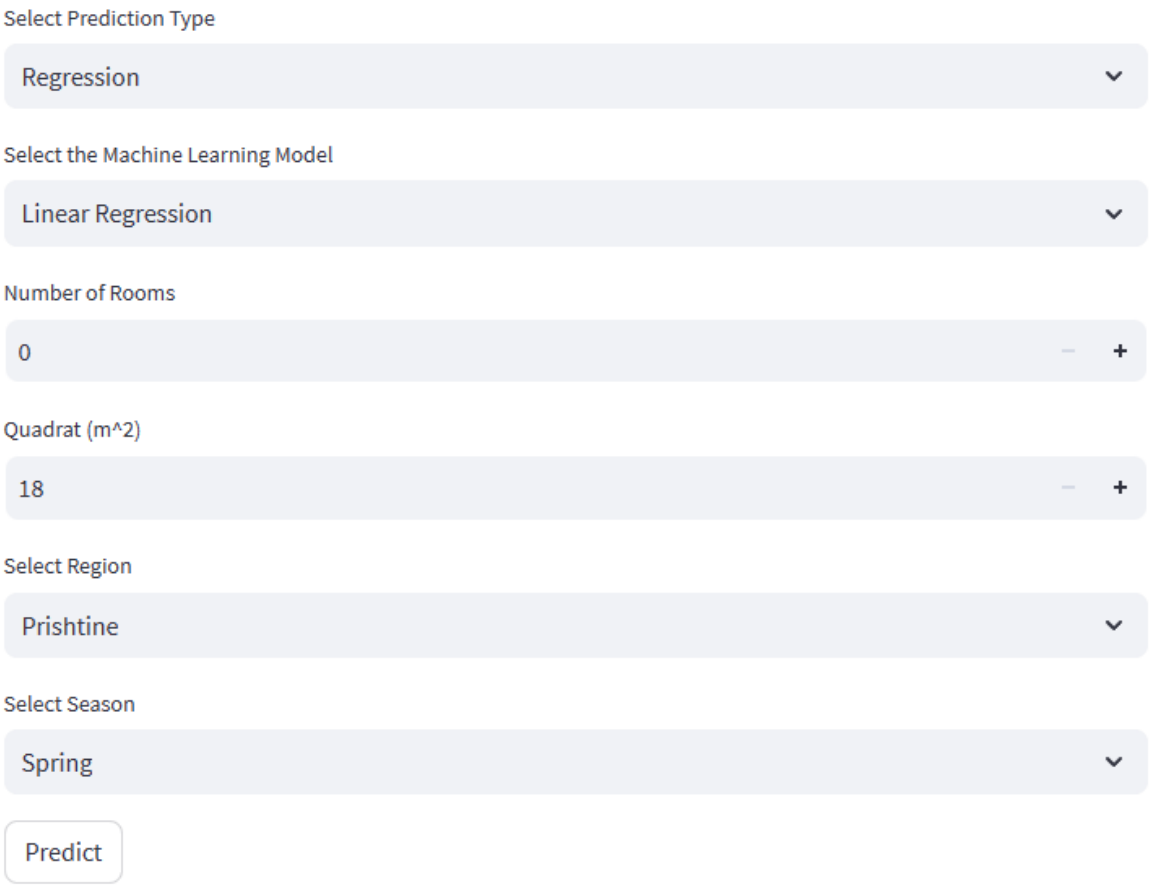

Figure 4. The initial view of the application.

This application is a simple web application build with the python streamlit library. The prediction type can be Regression or Classification. Based on the selection of the prediction type the machine learning models are shown. When Regression is choosen the Linear Regression, K-nn, Naive Bayes, Random Forest, Decision Tree algorithms are possible to be choosen, when Classification is choosen the Logistic Regression, K-nn, Naive Bayes, Random Forest, Decision Tree algorithms are possible to be choosen. The possible number of rooms that can be written are 1, 2, 3, 4, 5. The possible quadrat values are as follows: when the number of rooms is 0 the minimum quadrat that can be choosen is 18, and the maximum quadrat that can be choosen is 37. When the number of rooms is 1 the min is 37 and max is 56, when the number of rooms is 2 the min is 56 and max is 93, when the number of rooms is 3 the min is 93 and max is 140, when the number of rooms is 4 the min is 140 and max is 170, when the number of rooms is 5 the min is 170 and max is 500. The regions that can be selected are Prishtine or Other. The season that can be choosen are Spring, Summer, Autumn, and Winter.

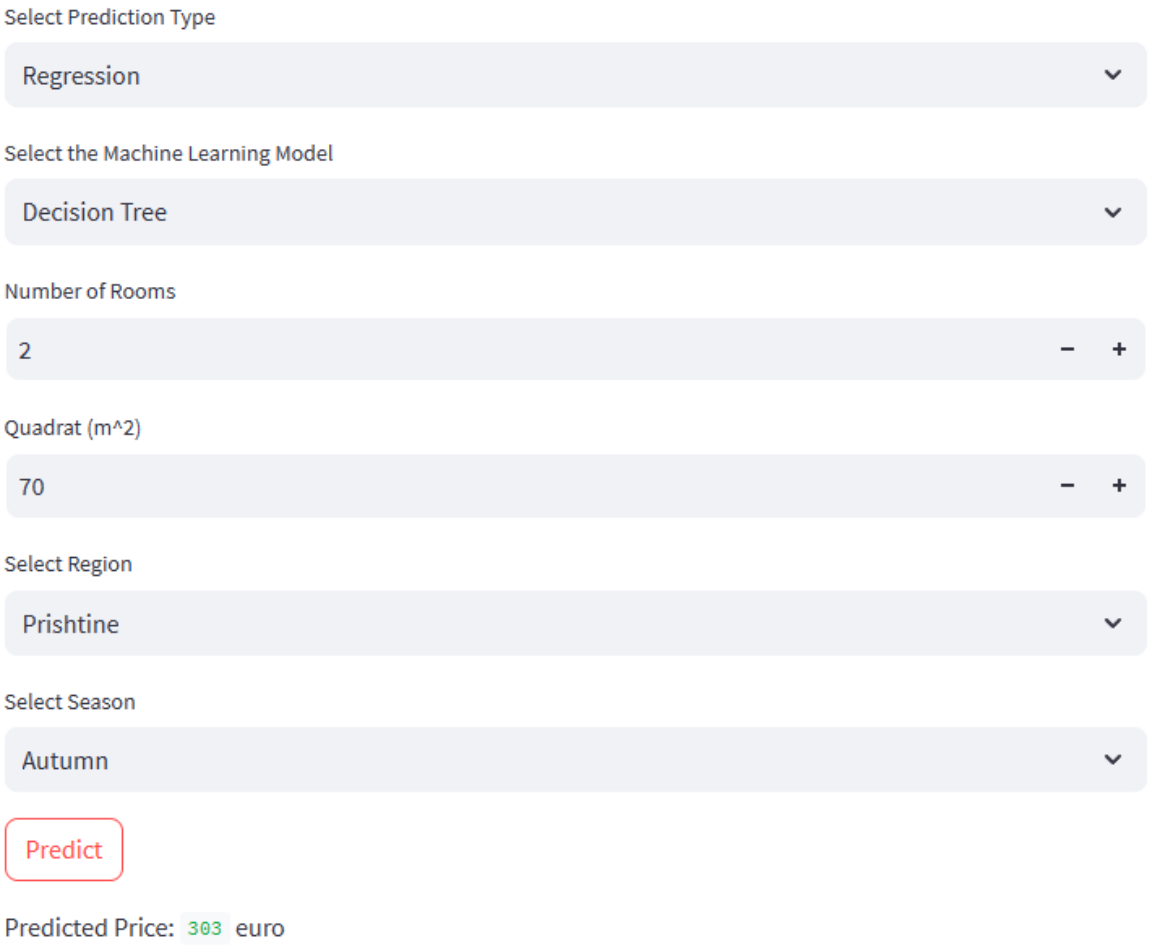

Figure 5. The view of the prediction after we choose some attribute values.

To get the prediction we click on the Predict button, after that the predicted price is going to be shown below.

This is a application that allows users to predict apartment rental prices in Kosovë based on various parameters. It uses pre-trained machine learning models which are loaded according to user's selection. The application uses the information that the user gives to perform a prediction using the selected model. The prediction results, either a price range (for classification) or a specific price (for regression), are displayed to the user. This provides an easy-to-use interface for predicting apartment renting prices in Kosovë utilizing various machine learning algorithms.

References

[1] Gjirafa. (Accessed: Day, Month, Year) [Online]. Available: https://gjirafa.com/Top/Patundshmeri?f=2&sh=Kosove&k=Banesa&llshp=Qira

[2] Python. (Accessed: Day, Month, Year) [Online]. Available: https://www.python.org/

[3] Streamlit. (Accessed: Day, Month, Year) [Online]. Available: https://streamlit.io/

[4] Joblib. (Accessed: Day, Month, Year) [Online]. Available: https://joblib.readthedocs.io/en/latest/

[5] Pandas. (Accessed: Day, Month, Year) [Online]. Available: https://pandas.pydata.org/

[6] Numpy. (Accessed: Day, Month, Year) [Online]. Available: https://numpy.org/

[7] Python Requests. (Accessed: Day, Month, Year) [Online]. Available: https://docs.python-requests.org/en/latest/

[8] Beautiful Soup. (Accessed: Day, Month, Year) [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[9] Python CSV. (Accessed: Day, Month, Year) [Online]. Available: https://docs.python.org/3/library/csv.html

[10] Python Time. (Accessed: Day, Month, Year) [Online]. Available: https://docs.python.org/3/library/time.html

[11] Matplotlib. (Accessed: Day, Month, Year) [Online]. Available: https://matplotlib.org/

[12] Seaborn. (Accessed: Day, Month, Year) [Online]. Available: https://seaborn.pydata.org/

[13] Scikit-learn. (Accessed: Day, Month, Year) [Online]. Available: https://scikit-learn.org/stable/

[14] Joblib/Threadpoolctl. (Accessed: Day, Month, Year) [Online]. Available: https://github.com/joblib/threadpoolctl

[15] U. Lajci, "Apartment price prediction kosove," GitHub. (Accessed: Day, Month, Year) [Online]. Available: https://github.com/uran-lajci/apartment-price-prediction-kosove

[16] Sailing Stone Travel, "Pristina Guide," (Accessed: Day, Month, Year) [Online]. Available: https://sailingstonetravel.com/pristina-guide/