DATA MINING

Dataset : Plots Prices In Pakistan

Group Members :

- M. Yousuf Noor
- Haider Hasan

Github Link: *https://github.com/yousufnoor5/dm_assignment_2*

Objective

This project aims to predict plot prices in Pakistan. Using a dataset of current plot sizes and prices from Zameen.com, a prominent real estate portal, we will develop a predictive model that can estimate the market value of plots based on their location, size, and other relevant factors. This model will serve as a valuable tool for potential buyers, sellers, and investors seeking insights into the dynamic Pakistani real estate market.

Introduction and Background of the Problem

The Pakistani real estate landscape is known for its constant fluctuations and complexities. Predicting plot prices can be a daunting task, influenced by a multitude of factors like regional dynamics, economic trends, government policies, and infrastructure development. This often leaves individuals investing in plots, be it for residential or commercial purposes, vulnerable to significant financial risks due to inaccurate price estimations.

This project addresses the problem of unpredictable plot prices in Pakistan by employing data mining techniques to extract valuable insights from Zameen.com's comprehensive dataset. By developing a predictive model, we aim to empower individuals with data-driven insights to make informed decisions, reduce financial risks, and navigate the intricate landscape of the Pakistani real estate market with greater confidence.

Data Collection

We couldn't find a dataset for plot prices in Pakistan, so we coded a Python script to scrape this information from zameen.com. It was a difficult task, because the zameen.com website uses a single-page application. To get all the information from the website, we had to automate Chrome with Selenium.

One challenge we faced was avoiding detection by bot detectors. We had to be careful to not get caught while collecting the data.

The scraper helped us to scrape the plots data quickly.

The main columns of our dataset :

- City Name
- Location Name
- Location Category
- Area Size (In square yards)
- Price

Data Preprocessing

Removal of Duplicate Data:

- We began by eliminating duplicate entries in the dataset. This was necessary as multiple ads for the same plots were found on Zameen.com.

Standardization of Area Units:

- To ensure consistency, we converted all area measurements to a common unit, square yards. This was essential as Zameen.com used various units, such as marla and kanal, depending on the city.

Encoding of Location Data:

- To enable the use of regression models, we needed to transform the text-based location data into a numerical format.
- While one-hot encoding was an option, it would have resulted in over 50 additional columns due to the 50+ unique locations across Pakistan.
- To reduce the dimensionality, we opted for binary encoding. This method assigns a unique binary number to each location, effectively representing it using only 8 columns.

One-Hot Encoding for City Names:

- To incorporate city names into the numerical model, we employed one-hot encoding. This technique creates a separate binary column for each city, indicating its presence (1) or absence (0) in each data point.

Grouped Location based on value:

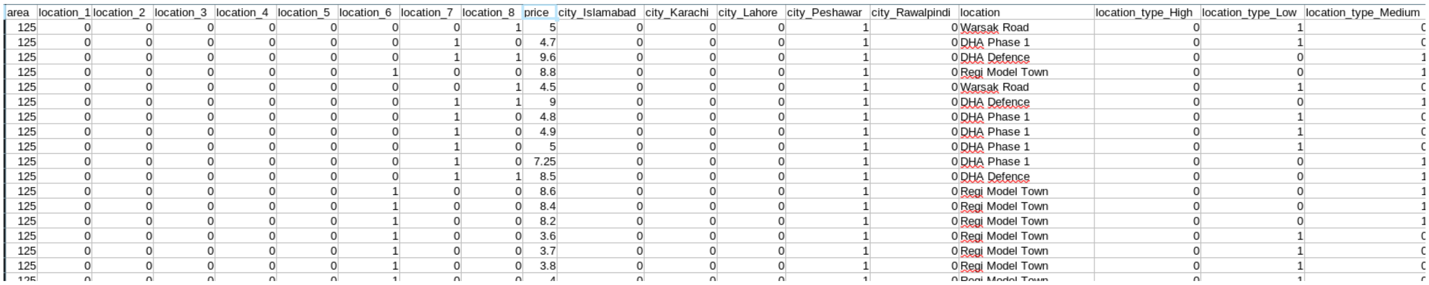
- We grouped location based on their value. For example, DHA plots prices were higher than north karachi. We grouped them in three category ‘low’, ‘medium’, ‘high’ and then converted them into 3 columns with 0/1 values depending on whether that location lies in that category. This technique is also known as one hot encoding.

Conversion of price:

- We converted the price column value into millions for better readibility

Dataset Look :

The location column with string values will be excluded on the rapid miner since we already binary encoded it.



area	location_1	location_2	location_3	location_4	location_5	location_6	location_7	location_8	price	city_Islamabad	city_Karachi	city_Lahore	city_Peshawar	city_Rawalpindi	location	location_type_High	location_type_Low	location_type_Medium
125	0	0	0	0	0	0	0	1	5	0	0	0	1	0	Warsak Road	0	1	0
125	0	0	0	0	0	0	1	0	4.7	0	0	0	1	0	DHA Phase 1	0	1	0
125	0	0	0	0	0	0	1	1	9.6	0	0	0	1	0	DHA Defence	0	0	1
125	0	0	0	0	0	1	0	0	8.8	0	0	0	1	0	Regi Model Town	0	0	1
125	0	0	0	0	0	0	0	1	4.5	0	0	0	1	0	Warsak Road	0	1	0
125	0	0	0	0	0	0	1	1	9	0	0	0	1	0	DHA Defence	0	0	1
125	0	0	0	0	0	0	1	0	4.8	0	0	0	1	0	DHA Phase 1	0	1	0
125	0	0	0	0	0	0	1	0	4.9	0	0	0	1	0	DHA Phase 1	0	1	0
125	0	0	0	0	0	0	1	0	5	0	0	0	1	0	DHA Phase 1	0	1	0
125	0	0	0	0	0	0	1	0	7.25	0	0	0	1	0	DHA Phase 1	0	0	1
125	0	0	0	0	0	0	1	1	8.5	0	0	0	1	0	DHA Defence	0	0	1
125	0	0	0	0	0	1	0	0	8.6	0	0	0	1	0	Regi Model Town	0	0	1
125	0	0	0	0	0	1	0	0	8.4	0	0	0	1	0	Regi Model Town	0	0	1
125	0	0	0	0	0	1	0	0	8.2	0	0	0	1	0	Regi Model Town	0	0	1
125	0	0	0	0	0	1	0	0	3.6	0	0	0	1	0	Regi Model Town	0	1	0
125	0	0	0	0	0	1	0	0	3.7	0	0	0	1	0	Regi Model Town	0	1	0
125	0	0	0	0	0	1	0	0	3.8	0	0	0	1	0	Regi Model Town	0	1	0
125	0	0	0	0	0	1	0	0	4	0	0	0	1	0	Regi Model Town	0	1	0

Modelling , Evaluation & Results

We split our dataset like this : the training dataset contains 80% of the total data and 20% data are in the testing dataset.

The maxmium value of price we have in our dataset is **104 M**
The minimum value of price we have in our dataset is **0.16 M**

Range = max - min = **103.84**

Our objective was to make sure that the root mean squared error should not be greator than this range and to get the least RMSE.

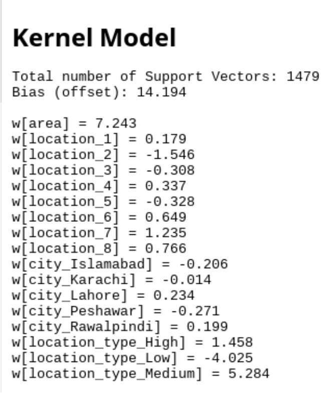
Support Vector Machine :

Kernel type : dot

The root mean squared error for SVM was : **7.952 +/- 0.000**

Its lesser than the RMSE but we need the least value, the another issue with the model was it was predicting some negative price values so we have to leave it.

Model picture :



Kernel Model

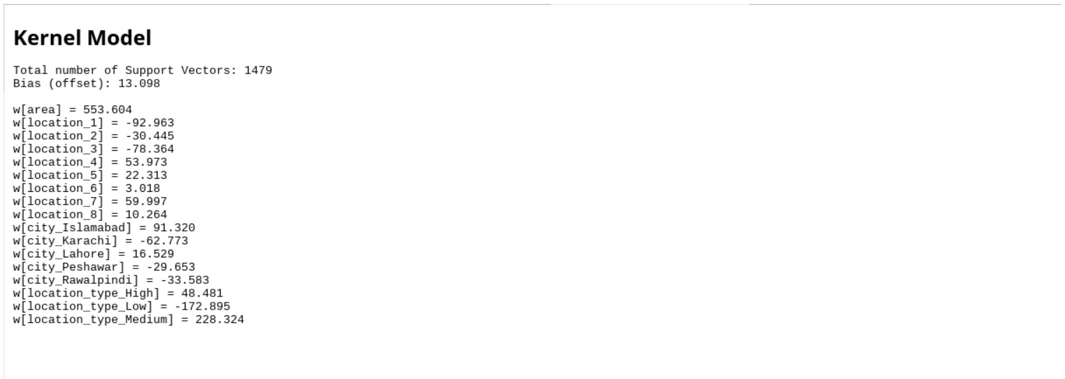
```
Total number of Support Vectors: 1479
Bias (offset): 14.194

w[area] = 7.243
w[location_1] = 0.179
w[location_2] = -1.546
w[location_3] = -0.308
w[location_4] = 0.337
w[location_5] = -0.328
w[location_6] = 0.649
w[location_7] = 1.235
w[location_8] = 0.766
w[city_Islamabad] = -0.206
w[city_Karachi] = -0.014
w[city_Lahore] = 0.234
w[city_Peshawar] = -0.271
w[city_Rawalpindi] = 0.199
w[location_type_High] = 1.458
w[location_type_Low] = -4.025
w[location_type_Medium] = 5.284
```

Kernel type : radial

The root mean squared error for SVM was : **7.822 +/- 0.000**

Model picture :



Kernel Model

```
Total number of Support Vectors: 1479
Bias (offset): 13.098

w[area] = 553.604
w[location_1] = -92.963
w[location_2] = -30.445
w[location_3] = -78.364
w[location_4] = 53.973
w[location_5] = 22.313
w[location_6] = 3.018
w[location_7] = 59.997
w[location_8] = 10.264
w[city_Islamabad] = 91.320
w[city_Karachi] = -62.773
w[city_Lahore] = 16.529
w[city_Peshawar] = -29.653
w[city_Rawalpindi] = -33.583
w[location_type_High] = 48.481
w[location_type_Low] = -172.895
w[location_type_Medium] = 228.324
```

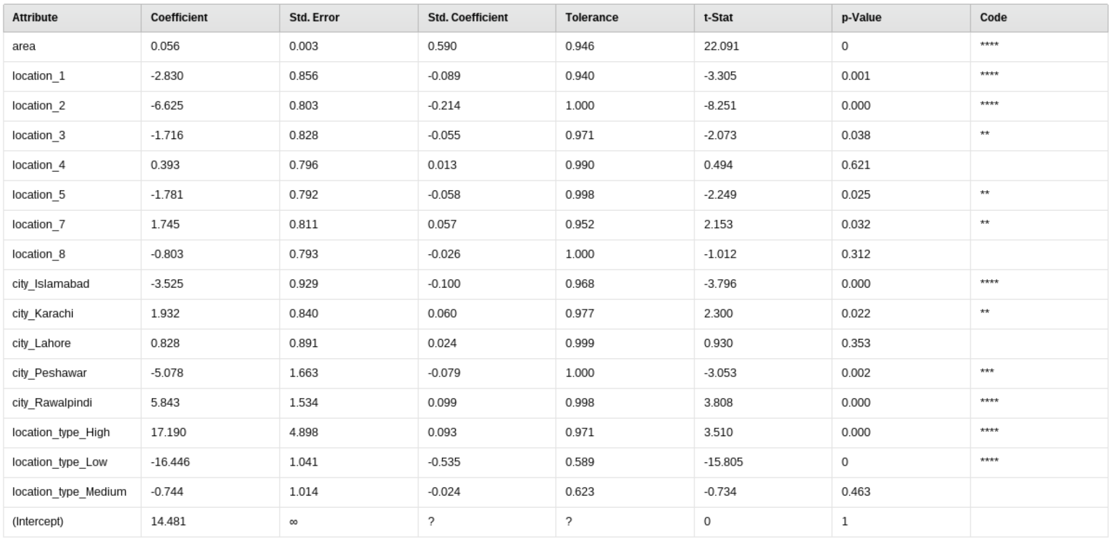
Its lesser than the RMSE and the dot kernel of SVM and it was also not predicting any negative values.

We also experimented with alternative kernels; however, their results were worse than those obtained with the radial kernel.

Linear Regression :

The root mean squared error for LR was : **7.291 +/- 0.000**

Model picture :



Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
area	0.056	0.003	0.590	0.946	22.091	0	****
location_1	-2.830	0.856	-0.089	0.940	-3.305	0.001	****
location_2	-6.625	0.803	-0.214	1.000	-8.251	0.000	****
location_3	-1.716	0.828	-0.055	0.971	-2.073	0.038	**
location_4	0.393	0.796	0.013	0.990	0.494	0.621	
location_5	-1.781	0.792	-0.058	0.998	-2.249	0.025	**
location_7	1.745	0.811	0.057	0.952	2.153	0.032	**
location_8	-0.803	0.793	-0.026	1.000	-1.012	0.312	
city_Islamabad	-3.525	0.929	-0.100	0.968	-3.796	0.000	****
city_Karachi	1.932	0.840	0.060	0.977	2.300	0.022	**
city_Lahore	0.828	0.891	0.024	0.999	0.930	0.353	
city_Peshawar	-5.078	1.663	-0.079	1.000	-3.053	0.002	***
city_Rawalpindi	5.843	1.534	0.099	0.998	3.808	0.000	****
location_type_High	17.190	4.898	0.093	0.971	3.510	0.000	****
location_type_Low	-16.446	1.041	-0.535	0.589	-15.805	0	****
location_type_Medium	-0.744	1.014	-0.024	0.623	-0.734	0.463	
(Intercept)	14.481	∞	?	?	0	1	

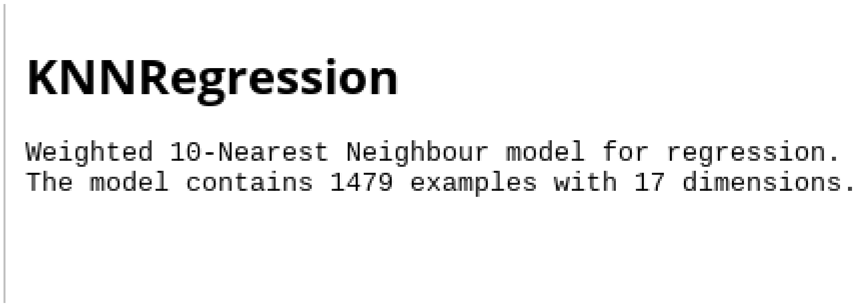
Its better than the previous model we used but still we want to check other models that might perform better with our dataset.

KNN (K-Nearest Neighbour) :

The root mean squared error was : **5.50 +/- 0.000** when **k = 5** and **5.304 +/- 0.000** when **k = 3**

It is relatively low than the range and better than all the previous models we tried.

Model picture :



KNNRegression

```
Weighted 10-Nearest Neighbour model for regression.
The model contains 1479 examples with 17 dimensions.
```

Random Forest :

To use random forest for regression task we had to change its criterion to **‘least_square’**

We tried different parameters of the random forest :

1-

Trees : 100
Maximal depth : 10

The root mean squared error was : **5.072 +/- 0.000**

2-

Trees : 100
Maximal depth : 15

The root mean squared error was : **5.008 +/- 0.000**

3-

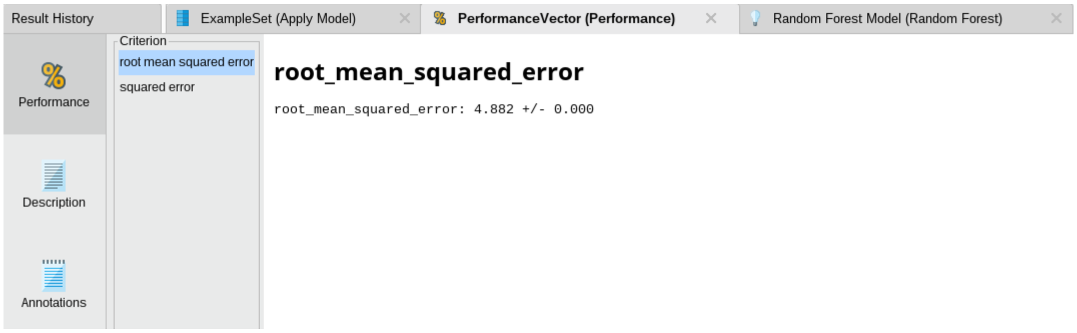
Trees : 200
Maximal depth : 15

The root mean squared error was : **4.958 +/- 0.000**

We tried with several different parameters but at the end these parameters gave us the best result :

Trees : 700
Maximal depth : 15

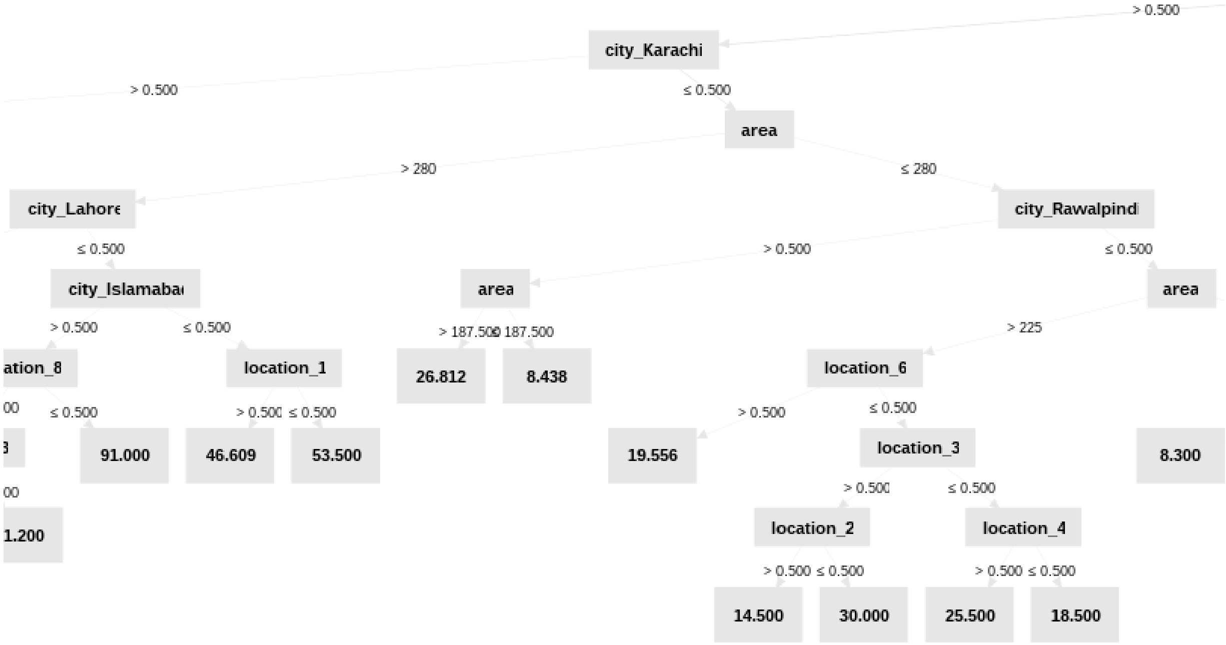
The root mean squared error was : **4.882 +/- 0.000**



root_mean_squared_error

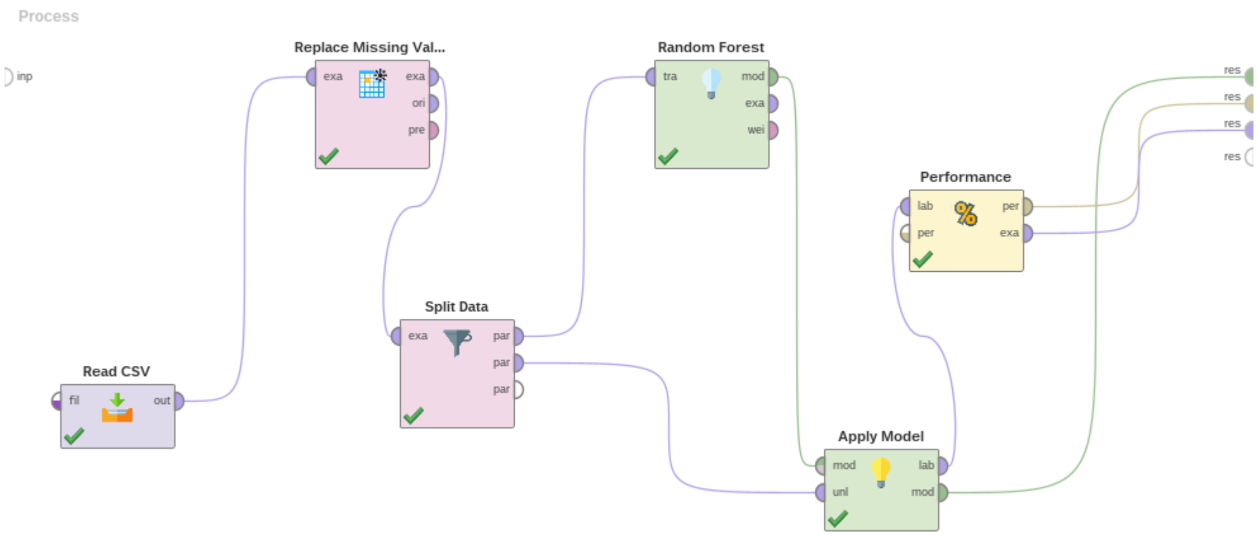
```
root_mean_squared_error: 4.882 +/- 0.000
```

Here is one of the model half picture , since the tree was very big. It is not possible to capture screenshot of the complete tree.



The predicted prices was also closely aligned with the actual prices in the test dataset.

Process Picture :



Conclusion:

In conclusion, our project successfully addressed the challenge of predicting plot prices in Pakistan, leveraging a dataset sourced from Zameen.com. The real estate market in Pakistan is known for its complexity, and our predictive model aimed to provide valuable insights for potential buyers, sellers, and investors.

The data collection process was challenging, requiring the development of a Python script to scrape information from Zameen.com, a single-page application. The use of Selenium for web automation and careful evasion of bot detectors enabled us to collect data efficiently.

While each model had its strengths and weaknesses, the overall objective was to minimize the RMSE and enhance the accuracy of price predictions. The developed predictive model, particularly based on KNN and Linear Regression, can serve as a valuable tool for individuals navigating the dynamic Pakistani real estate market. It provides a data-driven approach to estimate plot prices, thereby empowering stakeholders to make informed decisions, reduce financial risks, and contribute to a more transparent real estate ecosystem in Pakistan.