614CDS- Lab Project Report

Prediction Real Estates Prices in Riyadh

By

Basmah Misfer AlQahtani
443800986@kku.edu.sa

Maryam Mohmmad Alrashdi
443800993@kku.edu.sa

Submitted to

Associate Dr. Hamed AlQahtani

hsqahtani@kku.edu.sa

King Khalid University
College of Computer Science
Department of Information Systems
Semester II - Year 2022

Table of Contents:

List of Figures:

List of Tables:

1. Introduction:

To meet its goal of increasing Saudi citizen homeownership to 70% by 2030, the Saudi Ministry of Housing announced a proposal to build about 19,500 housing units under the "sakani" housing development program. The Ministry of Housing, in collaboration with the Real Estate Development Fund, initiated several programs to help residents get housing, and the Kingdom even implemented a rental price index to improve transparency and control in the residential real estate market.

Furthermore, due to the appearance of numerous advertisements on the websites of real estate apps such as (Haraj, Sohail, Aqar, and others), we discover that real estate prices have risen in various Riyadh areas.

It is worth noting here that one of the reasons for the rise in house prices in Riyadh is naturally the demand and supply of some lands in the high-end neighborhoods, and it has also decreased in many other areas, implying that the rise in prices was not only the master of the site in the estates of Riyadh, as many estates have fallen in price significantly.

In this project, we depended on estimating the prices of real estate in Riyadh based on their area and neighborhood, and we trained the models using a data set of home price announcements in Riyadh.

The number of rows was expected to be around ten thousand. We cleaned and processed the data, removing redundant and outlier data, and performed multiple manipulations to achieve high accuracy and correct prediction in several models, then compared the results.

2. Problem Definition:

In this project, we aim to predict house prices in Riyadh depending on the district where the house is located and the size of the house.

3. Dataset Description:

In this project, we used a database from Kaggle. This dataset contains over 10,000 records of Riyadh real estate advertising (for villas) for the year 2021. It is

The columns included in this dataset are:

- **District:** it contains all names of the district where the house is located.
- **House Price:** the price of each house in SAR.
- **House Size:** the area of each house.

4. Methods Used:

We tested numerous models in this project to guarantee that they perform well in this data set.

Here is a list of the models that were used:

1- Linear Regression.

2- Regularization models:

- Lasso.
- Ridge.
- ElasticNet.

3- Random Forest.

4- K-Nearest neighbors.

5. Data Settings and Preprocessing:

Data preprocessing is the process of converting raw data into a usable, intelligible format. Real-world or raw data is frequently inconsistently formatted, contains human mistakes, and may be incomplete. Such challenges are resolved through data preparation, which makes datasets more comprehensive and efficient for data analysis. It's an important step that can impact the performance of data mining and machine learning initiatives. It speeds up knowledge discovery from datasets and may eventually impact the performance of machine learning models.

In our project we make some preprocessing before applying ML algorithms they are:

1- There are no null values.
2- Remove duplicated rows.
3- Remove outliers.
4- Transform (district) column from categorical into numerical values by using (get_dummies) function.
5- Scale the training and test datasets.

6. Experiment Design:

1- We started experimenting with several models in this project because this data set had not yet been trained on any model.

We began by training the data on a linear regression model to predict future house price announcements. In the linear regression, we discovered that the model's performance is poor, as the value of R2 on the training set was higher than on the test set, where it was **-1.0869946287281051e+24**, indicating a poor performance for this model. We also did an experiment using the GridSearchCV function to adjust the parameters and found that the result became a little better.

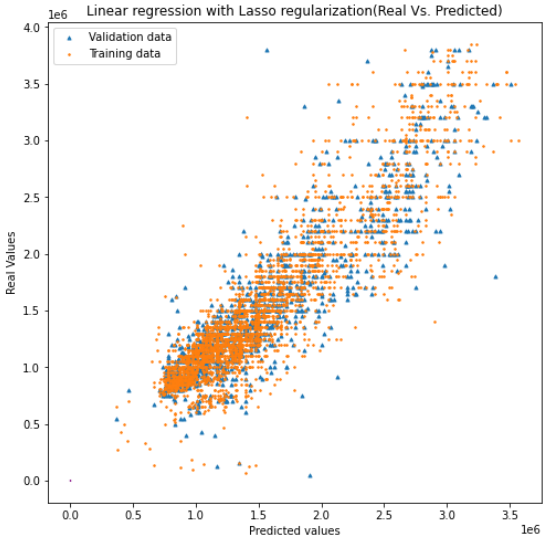
	Linear Regression without GridSearchCV	**Linear Regression with GridSearchCV**
***R^2* for the training dataset is:**	0.810436129493912	0.8104371027913506
***R^2*for the test dataset is:**	-3.601105619864604e+23	-1.5545441271348168e+21
The RMSE is:	4.088472019868982e+17	2.6862388599937784e+16
Average Error:	28687654146598384.0000	1310990670764723.5000

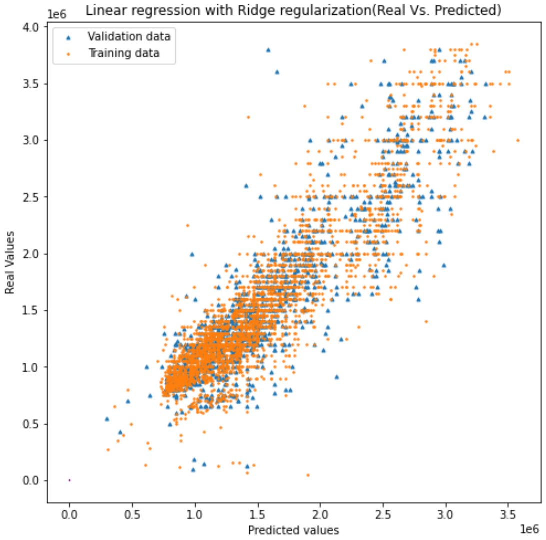
Table-1: Linear Regression Performance

2- We investigated the regularization models (Lasso, Ridge, and ElasticNet) after the linear regression experiment to get the optimum performance. The following table compares the performance of each system:

	Lasso	**Ridge**	**ElasticNet**
Accuracy	82.54%	82.53%	60.64%
***R^2* for the training dataset is:**	0.80974849	0.80967700	0.00012727
***R^2*for the test dataset is:**	0.78827422	0.78916621	-5.80215908
the RMSE is:	313494.532	312833.460	684601.884
Average Error:	218237.2558	218257.3475	525705.763

Table-2: Regularization models Performance





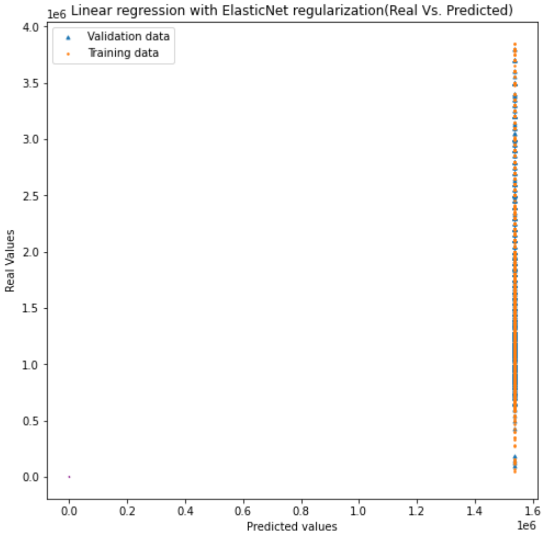


Figure-1: Real Vs. Predicted values in Lasso, Ridge, and ElasticNet

Looking at prior models, we discovered that ridge and lasso have equal prediction accuracy, however () perform poorly on both the training and test sets. When we applied the important features for Lasso picked 102 features such as (Muhammadiyah, Hiten, Arqah, and Almalqa) and eliminated the other 14 features.

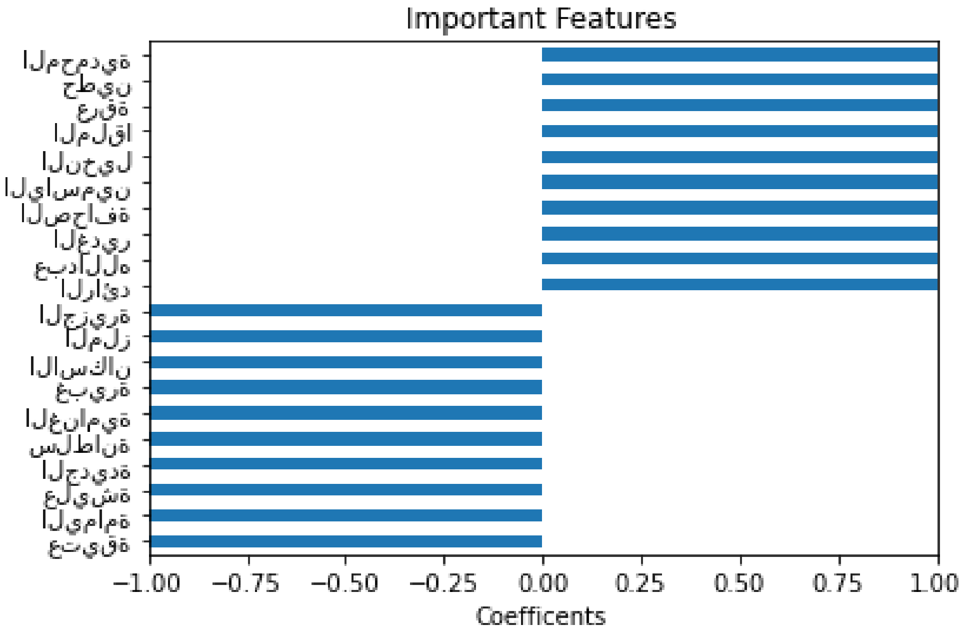


Figure-2: Important features in Lasso

The important features for Ridge picked 108 features such as (Muhammadiyah, Hiten, Arqah, and Al Nakheel) and eliminated the other 8 features.

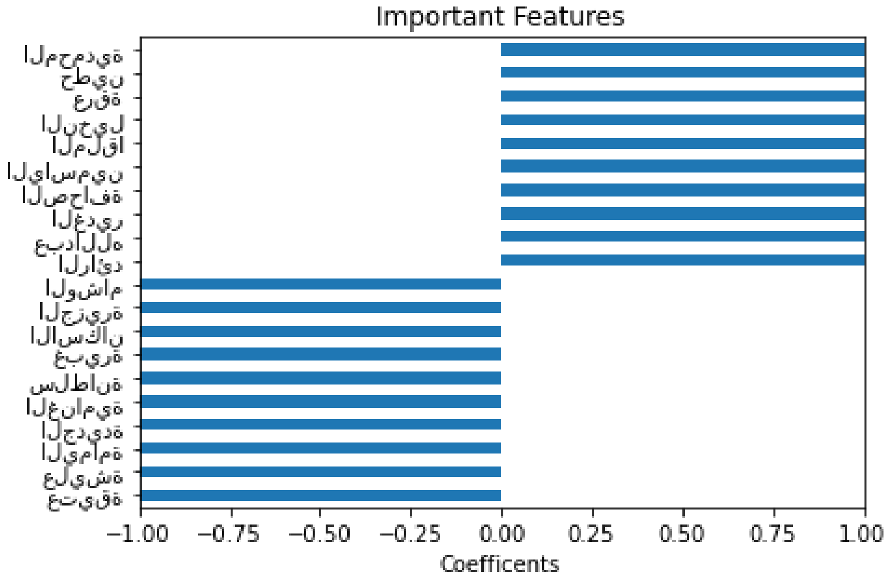


Figure-3: Important features in Ridge

The important features for ElasticNet picked 107 features such as (AL Qadeer, Muhammadiyah, Hiten, Arqah, and Alsahafah) and eliminated the other 9 features.

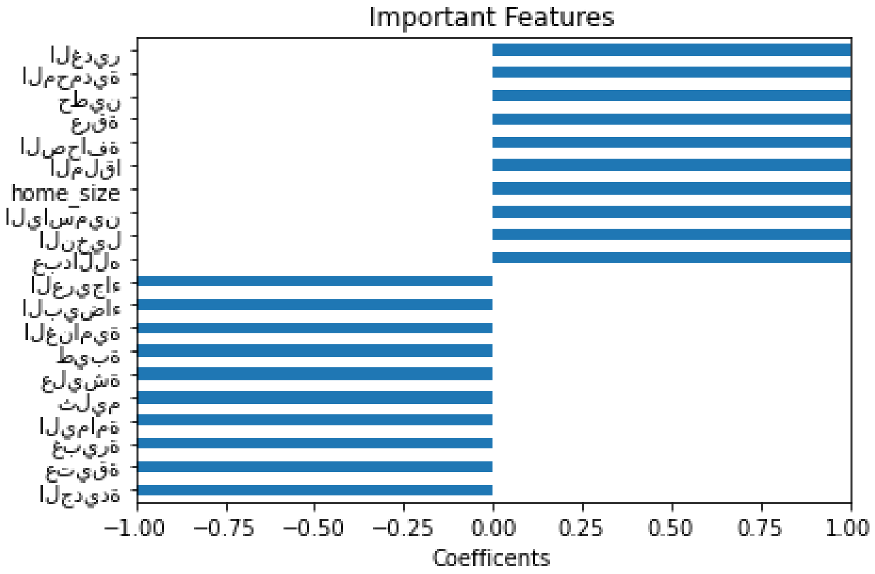


Figure-4: Important features in ElasticNet

3- We utilize Random Forest to predict the house prices and we found that the $\boldsymbol{R^2}$ is perform well on training and test sets (**0.86, 0.80** respectively). We use the parameters such as max_features, min_samples_split, and n_estimators to use it with the Gridsearch function to find the optimal fitting parameters. Then, use Out-of-Bag errors to calculate the average error using predictions from the trees that do not contain training observation in their respective bootstrap sample. This allows the RandomForestRegressor to be fit and validated whilst being trained. The plot below demonstrates how the OOB error can be measured at the addition of each new tree during training.

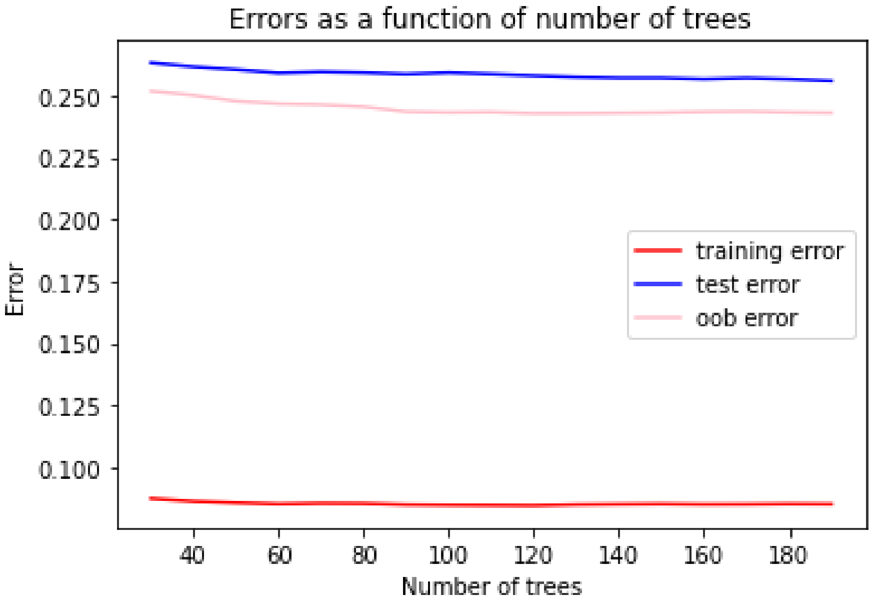


Figure-5: Errors as a function of the number of trees

4- Finally, we also experimented with this dataset to predict prices using K-nearest Neighbors (KNN) that do not perform well with using GridSearchCV and without using it the accuracy is better a little.

	KNN without GridSearchCV	**KNN with GridSearchCV**
Accuracy	62.77%	61.67%
R^2 for the training dataset is:	0.14034383	0.02860198
$R^2$for the test dataset is:	0.14519048	0.04612419
the RMSE is:	632999.185	668607.753
Average Error:	481663.9602	512699.0017

Table-3: KNN Performance

7. Conclusion:

At the end of this project, we can say that the random forest model is the best of the bunch, followed by the Lasso and Ridge models. We can also see that ElasticNet and the nearest neighbors model perform and estimate house values poorly. A summary of all models is provided below:

	LR	Lasso	Ridge	Elastic Net	RF	KNN
Accuracy	88232353223.34	82.54%	82.53%	60.64%	83.89%	62.77%
R^2 for the training dataset is:	0.81043710	0.80974849	0.80967700	0.00012727	0.86324143	0.14034383
R^2 for the test dataset is:	-1.554544127e+21	0.78827422	0.78916621	-5.80215908	0.80370058	0.14519048
the RMSE is:	2.6862388599937784e+16	313494.532	312833.460	684601.884	301857.941	632999.185
Average Error:	1310990670764723.5000	218237.2558	218257.3475	525705.763	202876.2464	481663.9602

Table-4: Summary of all models' Performance

8. REFERENCES:

- **Dataset:** MANSOUR. (2021). *Homes for sale in Riyadh*. Retrieved from Kaggle: https://www.kaggle.com
- Brownlee, J. (2020, October 12). *How to Develop LASSO Regression Models in Python*. Retrieved from machine learning mastery: https://machinelearningmastery.com/
- python, A.*MAPE - Mean Absolute Percentage Error in Python*. Retrieved from ask python: https://www.askpython.com
- readthedocs. *Neighbors Scalar Regression*. Retrieved from readthedocs: https://fda.readthedocs.io/
- scikit-learn. *Random Forest Regressor*. Retrieved from scikit-learn: https://scikit-learn.org/
- scikit-learn. *mean absolute error*. Retrieved from scikit-learn: https://scikit-learn.org/

9. Appendix: Program Code File

https://drive.google.com/file/d/1-AR3ppFWQwhxKlUrklqXprnRVoSP0gpt/view?usp=sharing