A Comparison of Lnear Regression, Regularization, and Machine Learning Algorithms in Predicting Apartments Prices in Riyadh,Saudi Arabia

Manal Anetallah Alsahafi

Introduction

The goal of this report is the analysis of major factors, influencing the performance of the linear regression (LR) algorithm in predicting the sales prices of apartments in the Saudi market. Riyadh is the capital of Saudi Arabia and the most populous city of the country. It has many branch municipalities and each one containing several districts, amounting to over 130 in total [1]. Therefore, this report aims to investigate the ability of the linear regression (LR) algorithm to predict the prices of the apartments in Riydh city by assessing the cost function and learning rate in the training phase and then compare the performance between our model and the polynomial regression with another regressor model.

Dataset Pre-process

The dataset that has been used in this work is a public dataset from from Kaggle called "Apartments in Riyadh Saudi Arabia". It has been collected and scraped from AQAR website which is considered the largest online real estate listing company in Kingdom of Saudi Arabia, allows agents and sellers to connect to renters and buyers all over the country [2]. The dataset contains only the data of apartments in Riyadh in 2022 that have been stayed on the AQAR website. It has around 6762 observations in it with 15 columns and containe mixual values between categorical and numeric. The sample dataset used illustrated in Table 1.

Table 1. Overview on the dataset

	district	latitude	longitude	area	age	num_bedrooms	num_livings	num_water_cycles	street_width	IsKetchen	IsFurnished	review	onMarket	IsRe
0	حي النظيم	24.800930	46.896890	225.0	9.0	3	0.0	2	15.0	1.0	0.0	5.00	17	Fals
1	حي القيحاء	24.687521	46.807558	130.0	12.0	3	1.0	2	30.0	1.0	0.0	4.33	5	Tru
2	حي الرمال	24.921463	46.806270	200.0	NaN	3	1.0	2	25.0	0.0	0.0	4.67	15	Tru
3	حي العقيق	24.780059	46.630602	120.0	0.0	1	1.0	1	34.0	1.0	0.0	4.17	165	Fals
4	حي التعاون	24.771793	46.698757	60.0	9.0	1	1.0	1	39.0	1.0	0.0	4.42	48	Fals
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6757	حي النرجس	24.866976	46.649873	180.0	0.0	3	1.0	2	18.0	1.0	0.0	4.31	33	Fals
6758	حي عنيزة	24.625145	46.735970	90.0	25.0	2	1.0	1	5.0	0.0	0.0	4.82	493	Fals
6759	حي البرموك	24.807284	46.767262	120.0	5.0	3	1.0	2	20.0	1.0	1.0	5.00	62	Fals
6760	حي النرجس	24.870188	46.650692	200.0	1.0	2	2.0	2	15.0	1.0	0.0	4.54	20	Fals
6761	حي النرجس	24.843407	46.679985	70.0	3.0	1	0.0	1	34.0	0.0	0.0	4.54	192	Fals

6762 rows × 15 columns

The reason behind collecting the dataset is to show how long apartments stayed on the market, As we are interested in the rental apartments in 2022 we restrained the dataset to this condition. Then we start cleaning the dataset from missing values by replacing NaN values with zero or dropping its rows and changing the datatype for int columns into float. Table 2 shows the dataset after preprocessing steps which decreased to 2600 examples.

Tabel 2: Clean Dataset

	district	latitude	longitude	area	age	num_bedrooms	num_livings	num_water_cycles	street_width	isKetchen	isFurnished	review	onMarket	isRe
1	حي البيضاء	24.687521	46.807558	130.0	12.0	3.0	1.0	2.0	30.0	1.0	0.0	4.33	5.0	Tru
5	حي الياسمين	24.814279	46.655360	170.0	3.0	3.0	1.0	2.0	35.0	0.0	0.0	4.47	7.0	Tru
9	حي النرجس	24.857649	46.655950	160.0	2.0	4.0	1.0	2.0	27.0	1.0	0.0	4.79	18.0	Tru
10	حي اشبيلية	24.796260	46.797633	150.0	1.0	1.0	0.0	1.0	39.0	1.0	0.0	4.29	17.0	Tru
11	حي الياسمين	24.804292	46.781654	8.0	0.0	2.0	1.0	1.0	29.0	1.0	0.0	4.40	36.0	Tru
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6747	حي القدس	24.754185	46.754499	75.0	3.0	2.0	1.0	1.0	22.0	1.0	0.0	4.78	37.0	Tru
6750	حي العارض	24.866398	46.619311	80.0	0.0	1.0	1.0	1.0	30.0	1.0	0.0	4.37	73.0	Tru
6753	حي الياسمين	24.834696	46.638251	175.0	2.0	4.0	1.0	5.0	19.0	1.0	0.0	4.31	18.0	Tru
6755	حي القيروان	24.823222	46.589722	175.0	0.0	3.0	1.0	2.0	30.0	1.0	0.0	3.74	39.0	Tru
6756	حي الياسمين	24.807219	46.767432	140.0	5.0	3.0	1.0	2.0	20.0	0.0	0.0	4.29	59.0	Tru

2600 rows × 15 columns

Exploratory Data Analysis (EDA)

“EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset” [3]. We applied different descriptive statistics for numerical data and creating various graphical representations to understand the data better. First to check the type for every data we call info() function as we had change all dataset into folat, we can calaute the sumery statistics using describe() function. see figure 1.

Fingure 1: The Descriptive Statistics

```
print(data.describe().round(2).T)
```

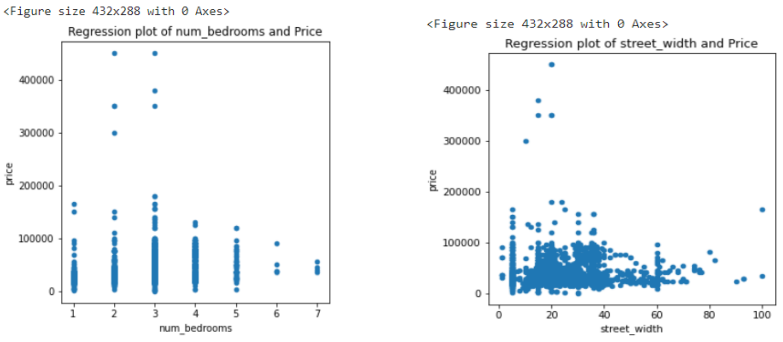
```
                  count       mean        std    min        25%        50%  \
area             2600.0     179.84     441.69    1.0     100.00     150.00
age              2600.0       2.65       4.17    0.0       0.00       1.00
num_bedrooms     2600.0       2.65       1.07    1.0       2.00       3.00
num_livings      2600.0       1.00       0.42    0.0       1.00       1.00
num_water_cycles 2600.0       2.09       0.87    1.0       1.00       2.00
street_width     2600.0      23.83      12.93    1.0      15.00      20.00
IsKetchen        2600.0       0.86       0.34    0.0       1.00       1.00
IsFurnished      2600.0       0.06       0.23    0.0       0.00       0.00
review           2600.0       4.12       1.03    0.0       4.14       4.36
price            2600.0   41531.47   28492.61   75.0   25000.00   35000.00

                     75%       max
area              190.00   18000.0
age                 4.00      36.0
num_bedrooms        3.00       7.0
num_livings         1.00       4.0
num_water_cycles    3.00       5.0
street_width       30.00     100.0
IsKetchen           1.00       1.0
IsFurnished         0.00       1.0
review              4.52       5.0
price           50000.00  450000.0
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2600 entries, 1 to 6756
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   area              2600 non-null   float64
 1   age               2600 non-null   float64
 2   num_bedrooms      2600 non-null   float64
 3   num_livings       2600 non-null   float64
 4   num_water_cycles  2600 non-null   float64
 5   street_width      2600 non-null   float64
 6   IsKetchen         2600 non-null   float64
 7   IsFurnished       2600 non-null   float64
 8   review            2600 non-null   float64
 9   price             2600 non-null   float64
dtypes: float64(10)
memory usage: 223.4 KB
```

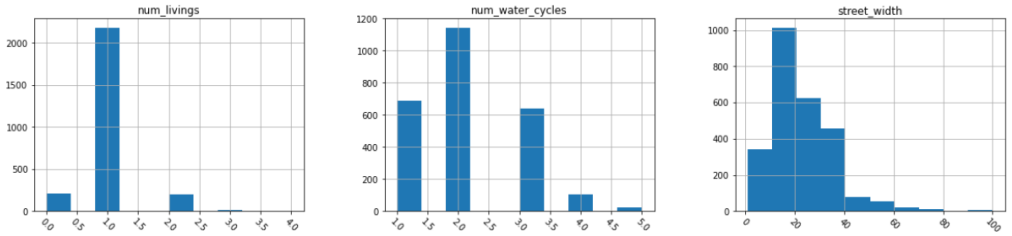
For more visualization, we used a Scatter plot to show the relationship between the 9 feature columns with respect to the target values in the “price” column in our dataset. Figures2 show the example of the shape of the relationship between the number of bedrooms and street width with respect to price.

Figure 2: Scatter plot



In addthion we plot the distribution of each feature to better understand the content of our data using histogram plots. Figures 3 shows the example of this kind of graphical representation.

Figures 3: Example of histogram plots



Feature engineering plays a key role in Machine learning and data mining algorithms; the quality of results of those algorithms largely depends on the quality of the available features[4]. The most tasks in feature engineering are: feature transformation, feature generation and extraction, feature selection, automatic feature engineering, and feature analysis and evaluation[4]. In this work, we satrt in removing the outliers and scalling dataset after we split them by using a ratio of 70:30, 70% training data, and 30% test data see figures 4.

Figures 4: The dataset after Scaling

	area	age	num_bedrooms	num_livings	num_water_cycles	street_width	IsKetchen	IsFurnished	review	price
0	-0.277913	3.400163	0.354227	0.008955	-0.081142	0.581712	0.409280	0.0	0.197755	-0.745918
1	0.068091	0.290411	0.354227	0.008955	-0.081142	1.010798	-2.443315	0.0	0.340516	0.013988
2	-0.018410	-0.055117	1.323624	0.008955	-0.081142	0.324260	0.409280	0.0	0.666825	0.013988
3	-0.104911	-0.400645	-1.584567	-2.573324	-1.320842	1.354066	0.409280	0.0	0.156967	-1.049881
4	-1.333224	-0.746173	-0.615170	0.008955	-1.320842	0.495895	0.409280	0.0	0.269135	-0.644598
...	...	...	...	...	...	...	...	...	...	...
2302	0.500595	-0.746173	-0.615170	2.591234	-0.081142	-0.276460	0.409280	0.0	0.697416	-0.239314
2303	-0.753668	0.290411	-0.615170	0.008955	-1.320842	-0.104826	0.409280	0.0	0.656628	-0.644598
2304	-0.710417	-0.746173	-1.584567	0.008955	-1.320842	0.581712	0.409280	0.0	0.238544	-0.492616
2305	0.111341	-0.746173	0.354227	0.008955	-0.081142	0.581712	0.409280	0.0	-0.403878	1.787101
2306	-0.191412	0.981467	0.354227	0.008955	-0.081142	-0.276460	-2.443315	0.0	0.156967	-0.695258

To fast the learning of our model and improve the accuracy we used Pearson's Correlation as a selection features method as it deals with the linear dependency between two continuous variables X and Y. In the pairs of features that are strongly correlated with each other, one of them should be removed to help a machine learning model to be more

generalized and interpretable [6]. We chose the value of the correlation between variables greater than 0.6 to be removed. The number of bathrooms in the apartment was the correlation column see Figure 5 after we remove it from the dataset.

Figure 5: The dataset after drop the correlation column

	Ones	area	age	num_bedrooms	num_livings	street_width	IsKetchen	IsFurnished	review
666	1	-0.537415	-0.055117	-0.615170	-2.573324	-0.877180	0.409280	0.0	0.656628
1566	1	1.192603	-0.746173	0.354227	0.008955	-0.448094	0.409280	0.0	0.371107
2078	1	-0.104911	2.709107	1.323624	0.008955	-0.705546	-2.443315	0.0	-0.046977
868	1	0.137292	-0.746173	1.323624	0.008955	-0.705546	0.409280	0.0	0.126375
200	1	-0.191412	-0.746173	0.354227	0.008955	-0.533912	0.409280	0.0	0.126375
...	...	...	...	...	...	...	...	...	...
1208	1	-0.450915	0.290411	0.354227	0.008955	-0.276460	0.409280	0.0	-0.199934
766	1	-0.243312	-0.746173	0.354227	0.008955	0.581712	0.409280	0.0	0.299727
1296	1	-0.364414	-0.055117	-0.615170	0.008955	1.439884	0.409280	0.0	0.177361
2267	1	2.057612	-0.055117	0.354227	0.008955	-0.276460	0.409280	0.0	-0.148948
154	1	-0.139511	-0.746173	0.354227	2.591234	-1.563718	0.409280	0.0	-4.217618

Methodology

In this report, we investigated the steps taken to achieve the desired output by implementing six different experiments. Building our model from scratch and using the gradient descent method to predict the apartment's rent price. Our model has been tested with 7 learning rate values. The learning rate = 0.1 was the best fitting for the model. In addition, we used a regularized term in our linear regression model to overcome the problem of overfitting. Two evaluation matrices were chosen which are Mean Square Error (MSE) and R Square (R2). MSE is simply the mean of the squared differences between predicted y and actual y. It was chosen because replay as a cost function for models that were built-in library in comperning with our model. R2 was used where there is a gap in the number of explanatory variables[7].

Results

The first investigation is about the impact of altering the cost function in linear regression. There was a huge difference between the two cost functions. In the first cost

function, we compute the summation of the squared differences between predicted y and actual y and divide by the number 2 but in the second cost function, we divide into 2 and multiply the length of X examples. The result of the two function shown in figure 6.

Figure 6: Rest of Cost Functions

```
print("FirstCost Function= ", FirstCost(X,y, theta))
print("SecondCost Function= ", SecondCost(X, y, theta))
```

```
FirstCost Function=  815.5392543509256
SecondCost Function=  0.5052907399943777
```

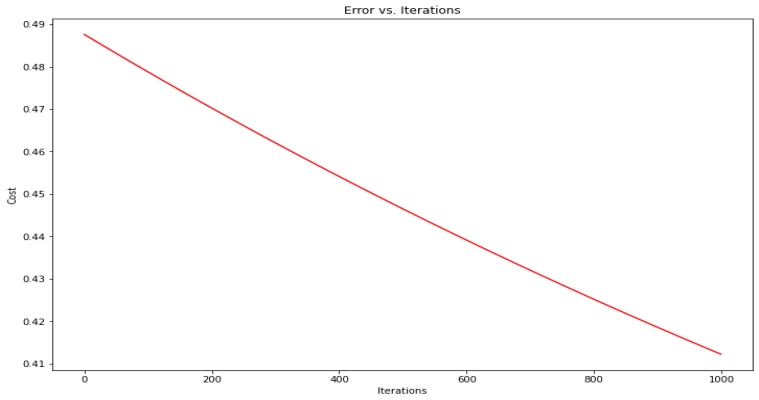
The second investigation is about the impact of learning rate on the training process where the α = {0.0001, 0.001, 0.01, 0.1, 1, 10, 100}. We fixed the iteration value at 1000. As we can see when we increase the value of the learning rate the time of learning and the error increase but the accuracy decreases and vers ver. In table 3 we compare the error and accuracy in the testing phase.

Tabel3: Comperion of Accuracy and Error

α	**0.0001**	**0.001**	**0.01**	**0.1 (Best)**	**1**	**10**	**100**
Error	0.41226406821475 56	0.2327641058148288 7	0.20890273157874 362	0.208798990902034 83	1.72628616457730 92e+120	nan	nan
Accuracy R2	0.98674332882366 03	0.6754811244235718	0.47508228034826 483	0.471735902319111 9	-3.3483117016260 217e+120	nan	nan

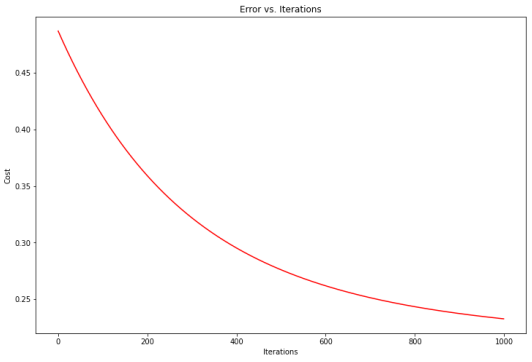
The plots of gradient against iteration below have a better visualization. In the first attempt, the learning rate was very small (α = 0.0001) and the gradient descent did not converge even after 1000 iterations and the cost is high. See figure 7.

Figure 7: First Attempt



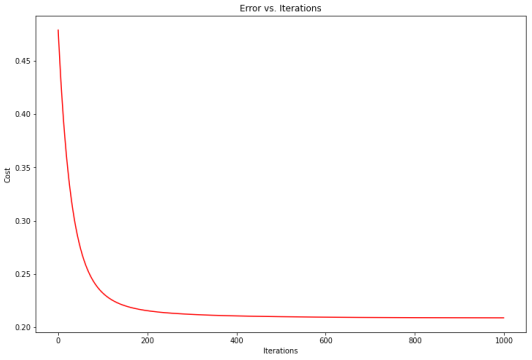
In the second attempt, the learning rate was also a small value (α = 0.001) even though the gradient descent converge before getting 1000 iterations but the speed to converge was slow. See figure 8.

Figure 8: Second Attempt



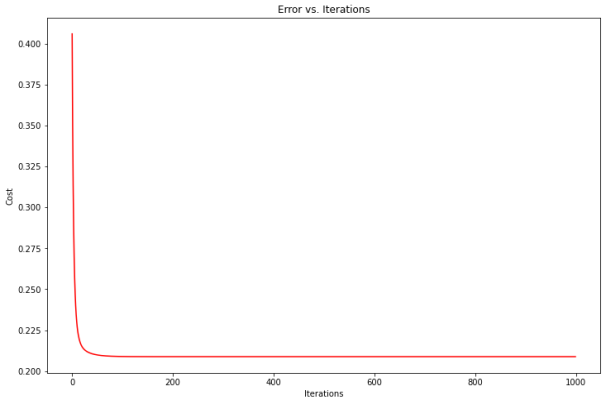
In the third attempt, the learning rate was the perfect one (α = 0.01) the gradient descent converge at 250 iterations and the gradient deos not move too high or too low. See figure 9.

Figure 9: Third Attempt



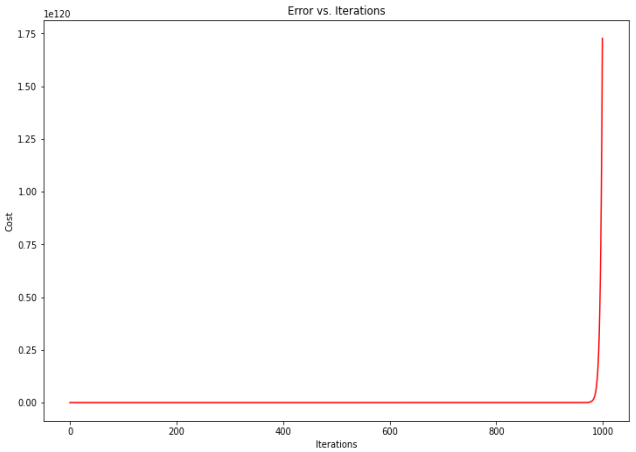
In the fourth attempt, the learning rate was bigger than the previous values ($\alpha = 0.1$) but the value very good and the gradient descent converge fast after 200 attempts but can be improved. See figure 10.

Figure 10: FourthAttempt



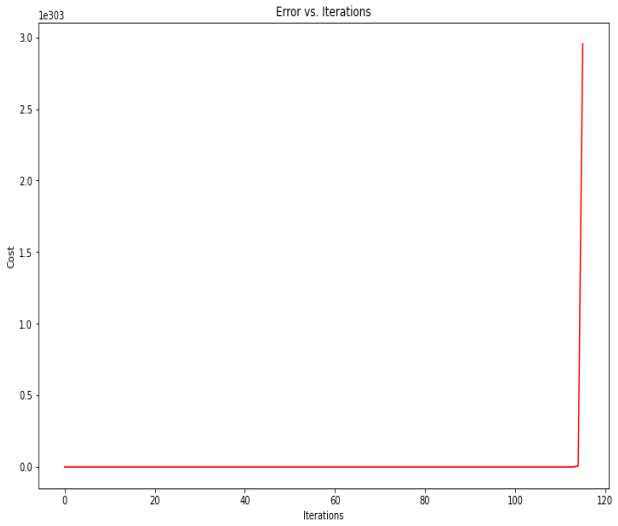
In the fifth attempt, the learning rate was a large value ($\alpha = 1$) and the gradient descent never converge since at end of the iterations error jumped to the maximum value. See figure 11.

Figure 11: Fifth Attempt



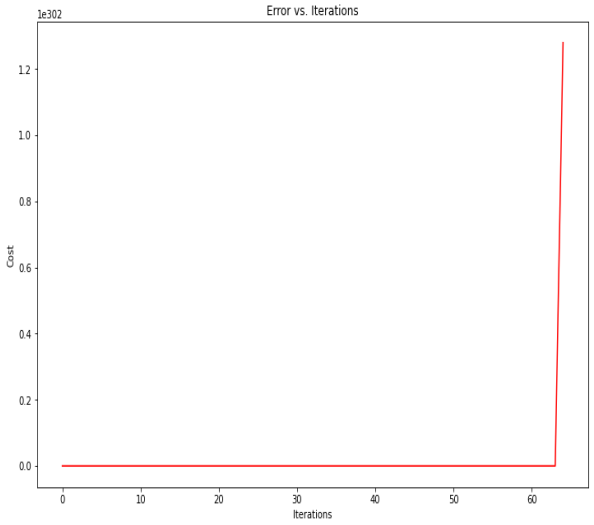
In the sixth attempt, the learning rate was very large value ($\alpha = 10$) tand the gradient descent never converge and befor get 1000 iterations the cost shoots up. See figure 12.

Figure 12: Second Attempt



In the seventh attempt, the learning rate was a huge value (α = 100) the gradient descent tand the gradient descent never converge and the cost shoots up. See figure 13.

Figure 13: Seventh Attempt



The thierd investigation is about the comparing our best model which has learning rate = 0.1 with sklearn model. The error in testing phase for our model above 21% and for the sklearn model above 42% while the accuracy of our model 47% and for sklearn model 57%.

The fourth investigation is about adding the regularization term into our linear regression. This regulation was added to gradient descent to improve our model and decrease the

overfitting problem since the error in training and testing are close to each other but equal to huge values. There was not a noticeable huge improvement since the rate was 1% in increasing accuracy and 1% in decreasing the error. So, the new error of our model = 20%, and the accuracy = is 48%.

The fifth investigation is about comparing our model with any other regressor model. In this stage, we choose the Ridge Regression model since it includes the regularization term called L2. The error of this model = 42% and the accuracy = 57%. The improvement was less than 1% from the stander linear regression of the sklearn model.

The sixth investigation is about using a polynomial regression of any degree. The accuracy o this model = is 86% where the error = is 12% in comparison with the rest model where the accuracy of our model with the regulzate term = is 48% and for the sklearn model = 57%.

Discussion and conclusion

This report proves many important insights, in the beginning, the altering in the cost function is very important to compute the correct error when modeling prediction. Choosing the right learning rate for gradient descent minimization is very expensive when we deal with huge datasets and plotting the cost is extremely recommended at this point. After all this effort in preprocessing steps, feature engineering, and improvements technique the result of our model was not good. however, the maxims accuracy and minums error was recorded by the polynomial regression model which leads us to conclude that the dataset used was non-linear nature.