CS 6375

ASSIGNMENT 1:

Linear Regression using Gradient Descent

Names of students in your group:

Rathang Rajpal - RXR210009

Yash Vijaynarayan Gupta - YXG210002

Number of free late days used: 0

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Note: Part1 - without libraries & Part2 - with sklearn libraries

Data loading and pre-processing

Data from UCI website Data Folder (also hosted on github in the code for consistency):

https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set

The data has been pre-processed using following steps to avoid noise that can spoil the model or even result in overfitting. The Real Estate Valuation dataset has a target of per unit area pricing value based on various attributes for which we have calculated the parameters or weights like tuning the knobs of a machine for desired finer results.

Removal of Nan values - No such values found to be removed in the dataset that can create noise.

Encoding of Categorical data - Encoding by label encoding was an option but was never required because the dataset lacks such categorical data.

Feature selection - Ensures that we set the dependency of target attribute on input attributes that are worth considering for model build-up. Feature selection has been discussed in detail in later parts of the report.

Data Normalization and Correlation

The normalization process holds importance of great value in this dataset as it brings down the scaling range of all the numerical data that would be considered in further procedures. Following formula was used to scale the data:

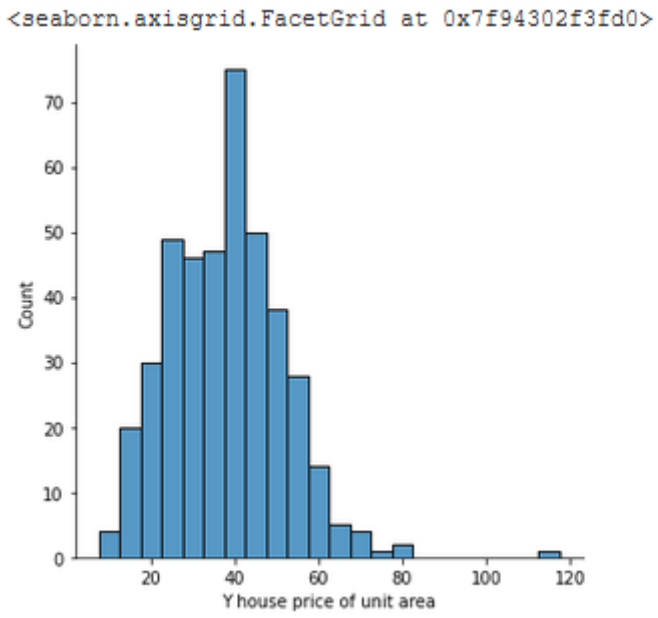
```
lambda x: (x-x.mean())/ x.std()
```

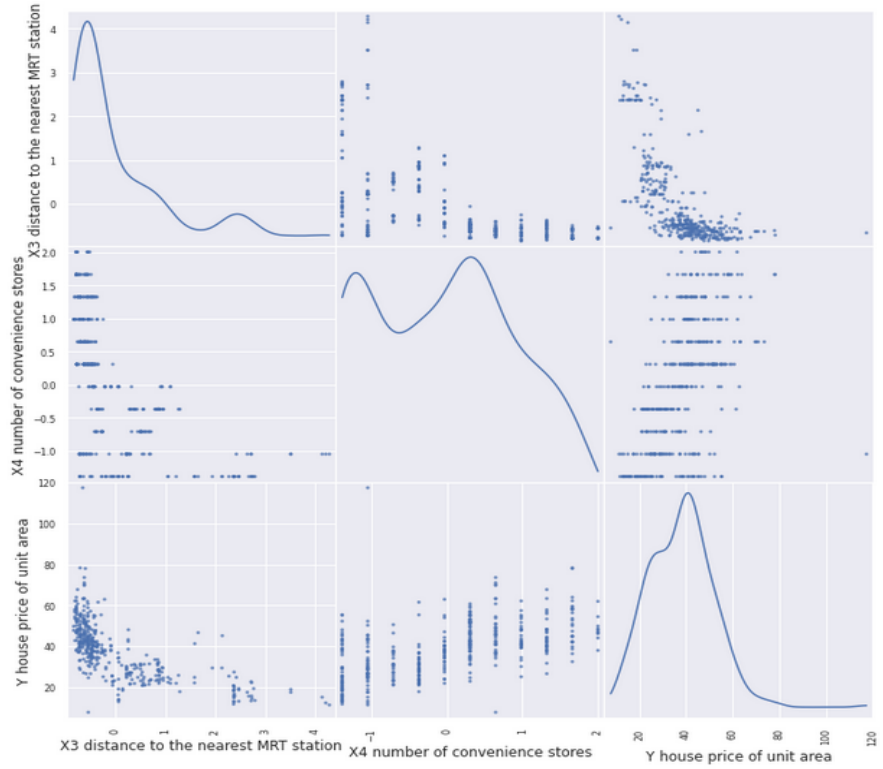
The normalized values do not disappoint the analysis done further to study the dataset for linear regression. It is evident by the Correlation matrix that only few values are considerable for modeling the regression and others have to be dropped. The heatmap further describes the same in color-coded fashion. The point to note here is that the dataset is not very good in terms of correlation so we skip the checking of multicollinearity which would aim to avoid the usage of those attributes that have good values of correlation amongst themselves. The point of focus in this assignment is to work on the SGD regressor with and without use of the sklearn library to check the difference of work.

Few plots for understanding the features

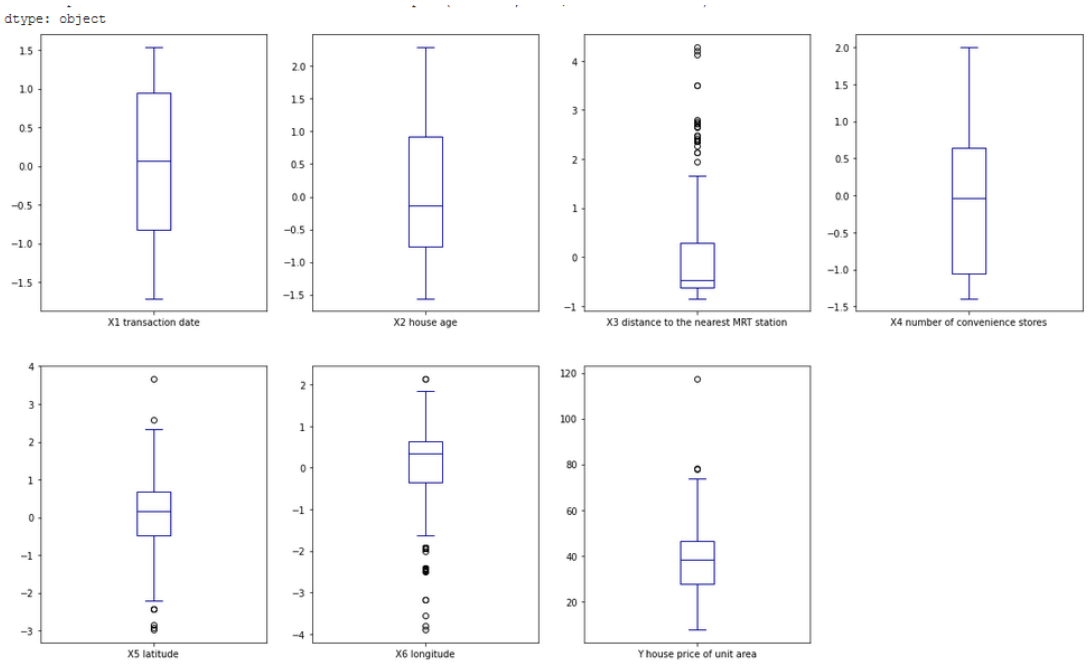
The features X3, X4 as described in the code are some of the best possible values for selection. Some of the plots that assist in understanding the same are below:

Histogram and pairplotting - It helps a lot to judge a dataset based on its target variable histogram. The pairplot of selected features allows us to analyze the distribution of data and whether it will be useful to be taken into account for model build-up.

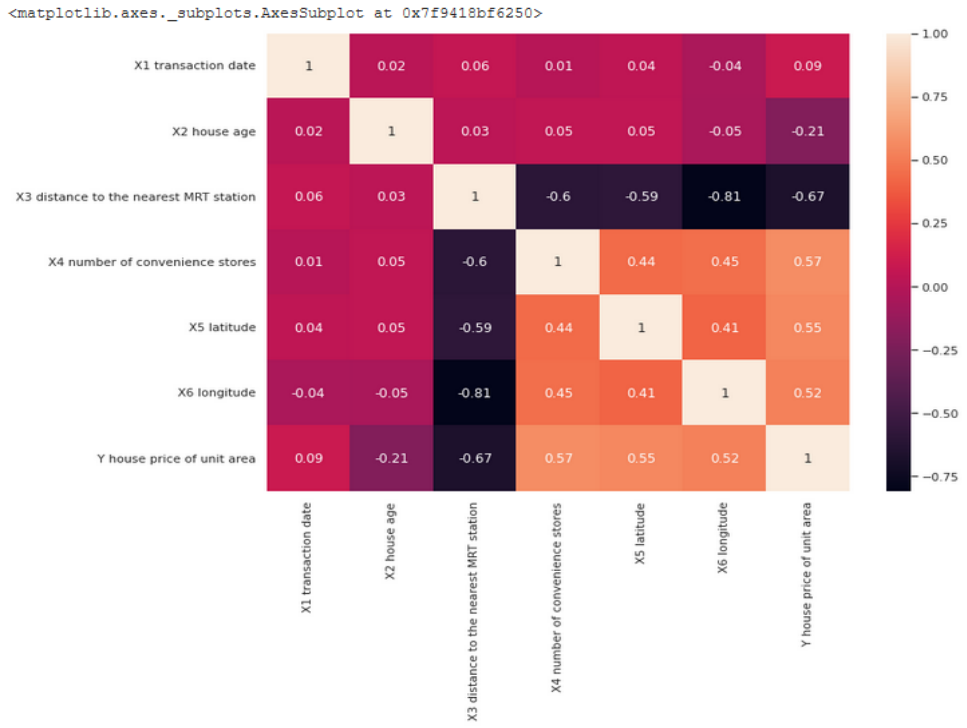




Box Plot - The distribution of data alongside its Mean, Median and quartile values makes it easy to grasp the importance of data or how much will it contribute to the model because the box-plot shows the outliers of the dataset.



Heat Map - The following heatmap color coding describes the correlation values as discussed before as well as the reasoning behind feature selection.



Note: The Pre-Processing, normalization of data and feature analysis discussed until now are common between Part1(without libraries) and Part2(by using libraries) of the Assignment. The Regression, Gradient Descent and related plots are discussed separately below.

Regression and Gradient descent(with and without libraries)

We have coded a regression class in part1 that has the fit and predict methods that tries to do the same task as the Linear Regression functions in sklearn libraries. The gradient descent and weight updation are two important functions where the evaluation plots of RMSE, R2 and Mean-absolute-error have been collected whenever it runs the fit function. The regression is fair as the values of input vectors have been standardized and the plots for the evaluation metrics alongside the change in learning rate and iterations have been clearly logged in this report further. The evaluation metric values are nearly the same for both the approaches of regression so it is satisfactory. The SGD approach in part 1 is by using following formula:

$$J = \frac{1}{2n}\sum_{i=1}^{n}(\text{Predicted Value} - \text{Actual Value})^2$$

where the sum is over all n data points.

Suppose the predicted model is $y = w_0 + w_1x_1$ where w_0 and w_1 are weights or coefficients for the bias and the first variable x_1 and let actual value be y_a

$$J = \frac{1}{2n}\sum_{i=1}^{n}[(w_0 + w_1x_1) - y_a]^2$$

To find the gradient w.r.t each of the w variables, we do the following:

$$\frac{\partial J}{\partial w_0} = \frac{1}{n}\sum_{i=1}^{n}[(w_0 + w_1x_1) - y_a]$$

$$\frac{\partial J}{\partial w_1} = \frac{1}{n}\sum_{i=1}^{n}x_1[(w_0 + w_1x_1) - y_a]$$

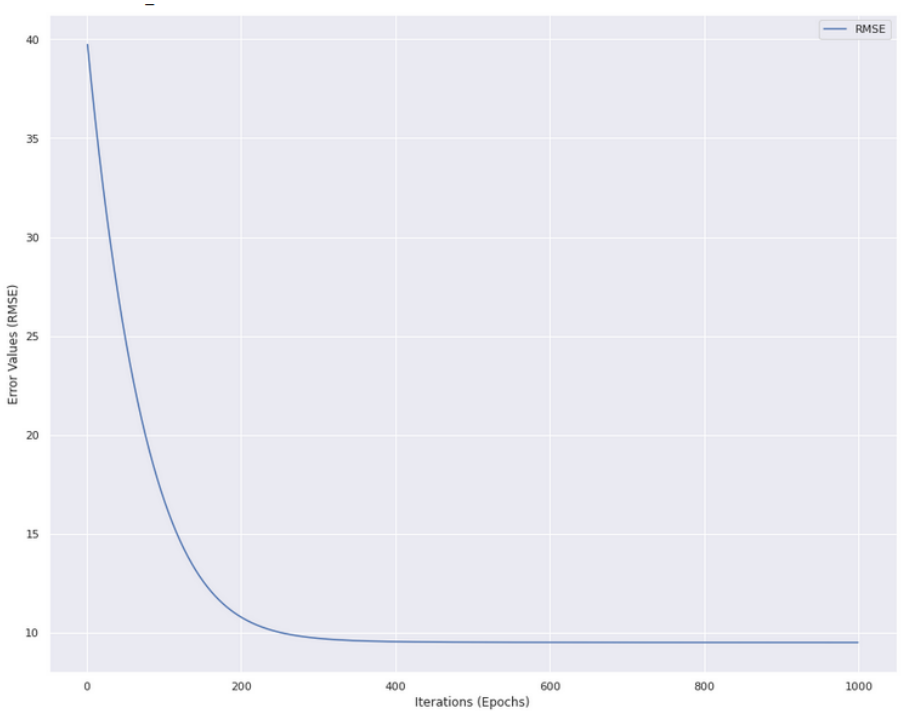
We can use above gradient functions to minimize the convex error function.

The value $[(w_0 + w_1x_1) - y_a]$ is called the residual i.e. difference between actual and predicted value for a data point.

The steep flow of error vs iterations shows how the gradient descent optimizes the weights or parameters with every passing iteration such that a minimum iteration is reached. We tried to log the evaluation metrics at low epoch counts to observe those changes closely so sometimes the library approach gave warnings that the descent could not be reached or it might have skipped a few steps. These metrics also vary depending on the learning rate variability and so the tabulation in the following report also throws light on it.

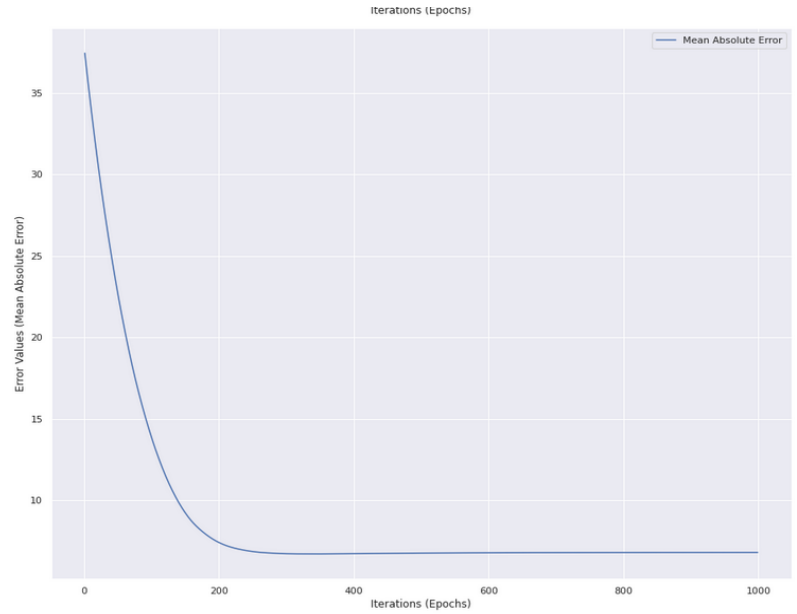
Plots: Iterations vs RMSE plot (vary w.r.t iterations and learning rate)

Part1: (Learning rate = 0.01 and n_Iterations = 1000)



Plot iteration vs Mean Absolute Error (vary w.r.t iterations and learning rate)

Part1: (Learning rate = 0.01 and n_Iterations = 1000)



Actual (Y_Test) vs Predicted(Y_prediction) values

Both show closeness in results which seems satisfactory for both parts of assignment

Part1:

Part2:

Log file creation

Part1:

Learning Rate = 0.1

S.No	Iterations	R2	MAE - Mean Absolute Err	RMSE - Root mean sq err
1	2000	0.4464770817116869	7.3796241088578585	10.212221559141124
2	5000	0.4464770817116869	7.3796241088578585	10.212221559141124
3	500	0.4464770819086259	7.379624107505563	10.21222155732441
4	1000	0.4464770817116869	7.3796241088578585	10.212221559141124
5	15000	0.4464770817116869	7.3796241088578585	10.212221559141124
6	50000	0.4464770817116869	7.3796241088578585	10.212221559141124
7	200	0.4464891154574484	7.379541330808719	10.212110550236368
8	20	0.28558843438636006	8.328381040727155	11.601834425596675
9	80	0.4472407481889422	7.374891715366637	10.205174495020282
10	150	0.4465518784873609	7.379104227643246	10.21153155424178

Learning Rate = 0.01

S.No	Iterations	R2	MAE - Mean Absolute Err	RMSE - Root mean sq err
1	2000	0.44649065037169944	7.379530746827014	10.212096390871737
2	5000	0.4464770819777282	7.379624107031069	10.21222155668696
3	500	0.44530812510419615	7.374177045392812	10.222999204922958
4	1000	0.44692808908621406	7.376090476914096	10.208060280848846
5	15000	0.44647708171168465	7.379624108857837	10.212221559141145
6	50000	0.44647708171168465	7.379624108857837	10.212221559141145
7	200	0.2583380814909366	8.541866312518607	11.82103259739345
8	20	-4.746055899445383	30.701804090073143	32.903140210631356
9	80	-1.1880275031343461	17.931695397643146	20.303869489366086
10	150	-0.005048492797513626	10.66223271125153	13.760871301515602

Learning Rate = 0.001

S.No	Iterations	R2	MAE - Mean Absolute Err	RMSE - Root mean sq err
1	2000	0.25554482033775605	8.564034443403829	11.843271975818805
2	5000	0.4452373239580021	7.374240505757654	10.223651618451017
3	500	-2.4606352104037392	23.451288178060484	25.53467507709088
4	1000	-0.6868746854645942	15.177324058803125	17.827639164373874
5	15000	0.4465595101353025	7.3790498582189405	10.211461149194413
6	50000	0.44647708198565395	7.379624106976643	10.212221556613846
7	200	-4.755452930798668	30.725661459463964	32.930033921438174
8	20	-6.980174481654465	36.35156583118489	38.775608185658164
9	80	-6.144905429006074	34.35677415693748	36.69024968304747
10	150	-5.293019933612364	32.168956599468096	34.43356419215205

Part 2:

Learning Rate = 0.1

S.No	Iterations	R2	MAE - Mean Absolute Err	RMSE - Root mean sq err
1	2000	0.4527270787000812	7.326963978822369	10.154403222244019
2	5000	0.4526744916196672	7.326115447038448	10.154891075352538
3	500	0.4523729549287966	7.321518751653379	10.157687994713406
4	1000	0.45279243220850396	7.320202459649578	10.153796901646139
5	15000	0.4523517336542223	7.323424978326362	10.157884804750141
6	50000	0.45253216989176714	7.321593080745129	10.156211283968137
7	200	0.4527432211529516	7.324588189734507	10.155436299966983
8	80	0.4524872598249037	7.322064336492137	10.156627844311114
9	20	0.45267296784914024	7.320438538942573	10.154905211102395
10	150	0.4523706307855546	7.324621276659321	10.157709549434882

Learning Rate = 0.01

S.No	Iterations	R2	MAE - Mean Absolute Err	RMSE - Root mean sq err
1	2000	0.4476468772186116	7.36789979665726	10.201424780134792
2	5000	0.4476568016149777	7.366011114196775	10.201333132750523
3	500	0.44783057789318925	7.369130531539149	10.199728252699677
4	1000	0.44711761236099945	7.367984415001062	10.206311113386509
5	15000	0.4473914685989455	7.370897483313226	10.203783081902065
6	50000	0.44777999523638556	7.371094868513606	10.200195425923946
7	200	0.4484187059370769	7.361226523379317	10.194294825760158
8	20	0.44830906863053266	7.3518926273094225	10.195307930722702
9	80	0.44817613198354145	7.369056603214013	10.196536198524752
10	150	0.4478439241140192	7.363012001909991	10.199604985598317

Learning Rate = 0.001

S.No	Iterations	R2	MAE - Mean Absolute Err	RMSE - Root mean sq err
1	2000	0.44735279060075805	7.374506903677439	10.20414016560201
2	5000	0.446577115596108	7.380235694534114	10.21129872982678
3	500	0.44708858906506077	7.371654304936435	10.20657899757591
4	1000	0.44693470050940765	7.375167921924154	10.207999267075715
5	15000	0.4469152456055583	7.374059803846093	10.208178806361559
6	50000	0.44676792543937505	7.380384690687458	10.20953824581137
7	200	0.44673916760742793	7.374915130253435	10.209803595908763
8	20	0.4473849616207959	7.374973164676395	10.203843156615296
9	80	0.4475627360361347	7.370124001613994	10.202201753435084
10	150	0.4467477273883468	7.382413608633957	10.209724615060203

Conclusion of assignment -

Are you satisfied that you have found the best solution?

We can conclude that best results are not very far apart from each other if we consider the results as evidenced by the Evaluation metrics from both the approaches (with and without libraries). The plots of these metrics against each passing iterations depict the gradient descent happening in the regression function in the Part1 of the assignment

Best values for the model were seen at:

Part 1:

Iterations: 1000

Learning Rate: 0.01

R2 value: 0.44692808908621406

Part 2:

Iterations: 200

Learning Rate: 0.1

R2 value: 0.4527432211529516

Are you satisfied that the package has found the best solution?

Yes we are satisfied with the results of the package and they are optimally better than our coded linear regression class and functions.

Sources used in assignment:

1. https://numpy.org/doc/stable/reference/index.html
2. https://pandas.pydata.org/docs/reference/index.html#api
3. https://matplotlib.org/stable/index.html