ML Assignment 1

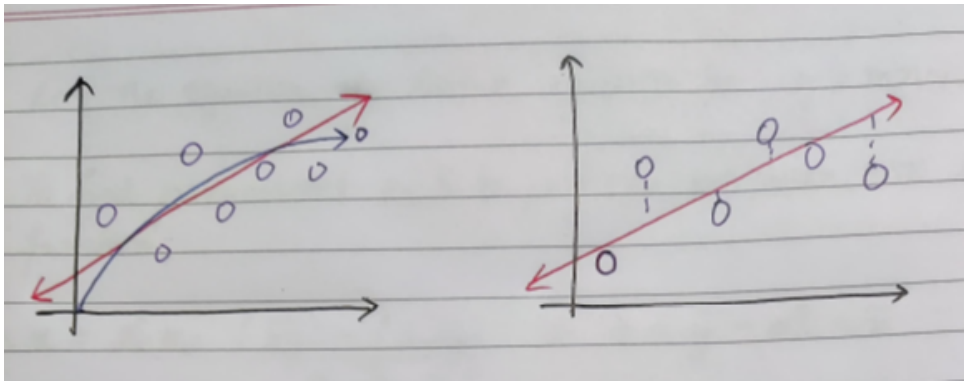
Saumil Lakra, 2021097

Q1:

(a) A model that has low complexity may not be able to capture the true relationship between features and output labels leading to underfitting leading to high bias and low variance. If we increase the complexity of the model by adding more features or include higher-order polynomial functions in the regression model, there is more chance that there will be overfitting on the training dataset because it can capture the true relationship between features and output labels really well. Such a model can lead to high variance on the testing dataset. This is the case of low bias and high variance.

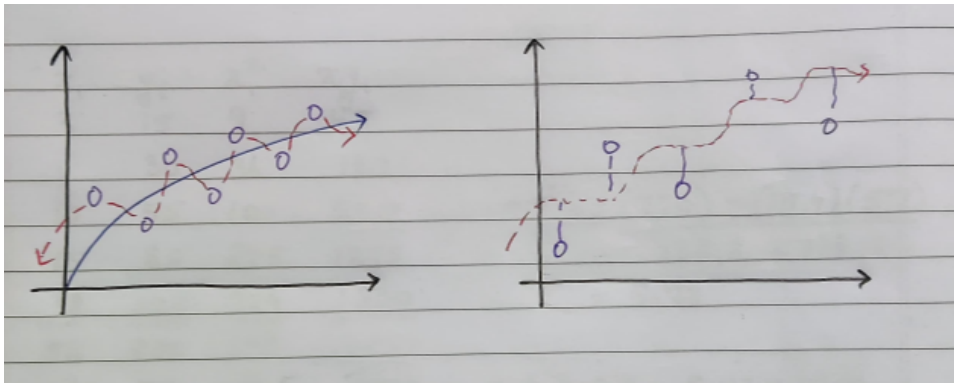
In both the graphs left graph is the training data and left one is testing data.

Underfitting (high bias and low variance):



Left graph: red line is the regression line. Since it is a simple linear polynomial, it fails to capture relationships between features and true labels leading to underfitting. (the red line is the same on both graphs.

Overfitting(low bias and high variance):



A complex (higher degree) polynomial is able to capture the relationship between features and labels in training data(left graph) but gives a high variance on test data(right graph). The red polynomial is same on both graphs.

(b):
TP=200, TN=730, FP=20, FN=50

(a) Precision (P) :

$$\frac{TP}{TP+FP} = \frac{200}{200+20} = \frac{200}{220} = 0.909$$

(b) Recall (R) :

$$\frac{TP}{TP+FN} = \frac{200}{200+50} = \frac{200}{250} = 0.8$$

(c) F1 Score:

$$\frac{TP}{TP+FP} = 2.\frac{P.R}{P+R} = 2X\frac{0.909X0.8}{0.909+0.8} = 0.851$$

(d) Accuracy:

$$\frac{TP+TN}{TP+TN+FP+FN} = \frac{TP}{TP+FP} = \frac{200+730}{200+730+20+50} = \frac{930}{1000} = 0.930$$

(c) :

Let the equation of linear regression be $y = mx + b$

To find parameters m & b, we can minimize MSE loss function,

$$m = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2} \quad \& \quad b = \bar{y} - m\bar{x}$$

	x_i	y_i	x_i^2	$x_i y_i$
	3	15	9	45
	6	30	36	180
	10	55	100	550
	15	85	225	1275
	18	100	324	1800
sum	52	285	694	3850
avg	10.4	57	138.8	770
	↑	↑	↑	↑
	$\bar{x}$	$\bar{y}$	$\overline{x^2}$	$\overline{xy}$

$$m = \frac{(770) - (10.4)(57)}{138.8 - 108.16} = 5.78$$

$$b = 57 - 5.78 \times 10.4 = -3.112$$

$$\Rightarrow y = 5.78x - 3.112$$

$x = 12$

$$y = 5.78 \times 12 - 3.112 = 66.248$$

(d).

Let the model f1 be $y = \sqrt{4x}$

Let the model f2 be $y = x$

Let the training data be:

	X	Y
0	0	0.000000
1	1	2.000000
2	3	3.464102
3	5	4.472136
4	4	4.000000
5	2	3.000000

As we can see, the f1 model has lower empirical risk than the f2 model for this training dataset.

Testing data:

	X	Y
0	10	10
1	11	11
2	12	13

If we use f1 then y(10) = $\sqrt{40}$ = 6.32

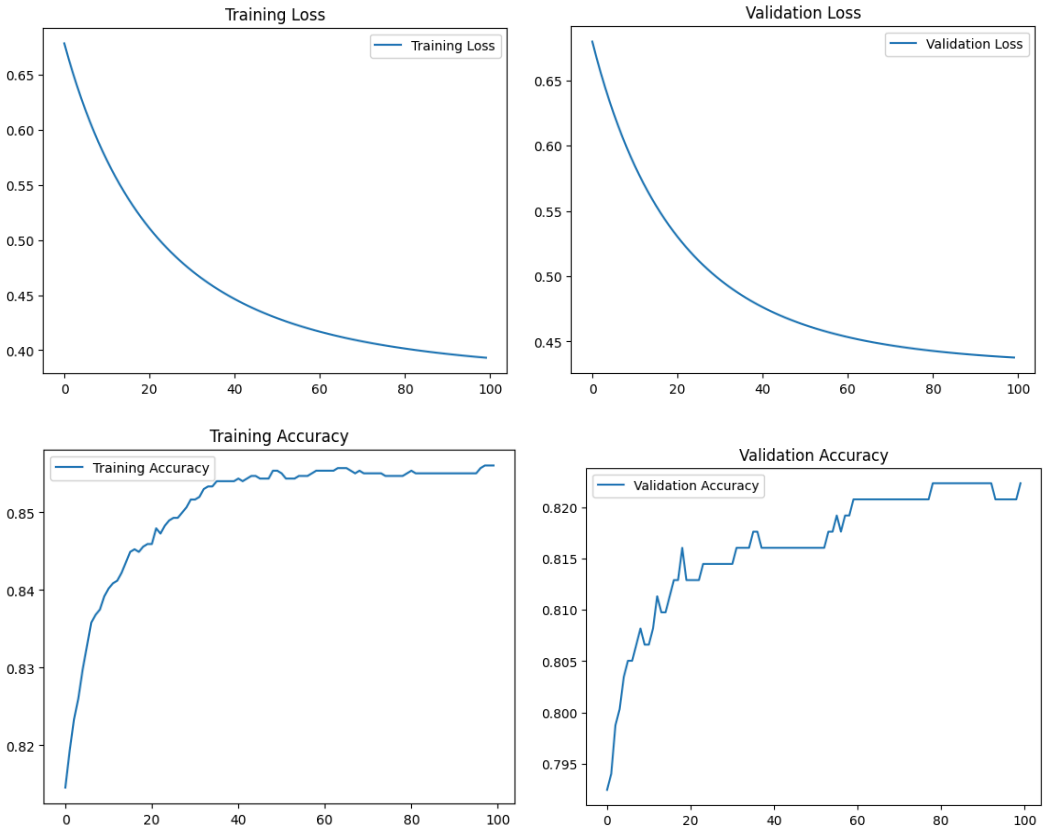
y(11) = $\sqrt{44}$ = 6.63

But if we use f2, we get accurate results.

Therefore even though f1 has lower empirical risk on the training set than f2, it can't necessarily generalize better than model f2. Model f1 has overfit on the training dataset.

Section B:

(a) Plots:



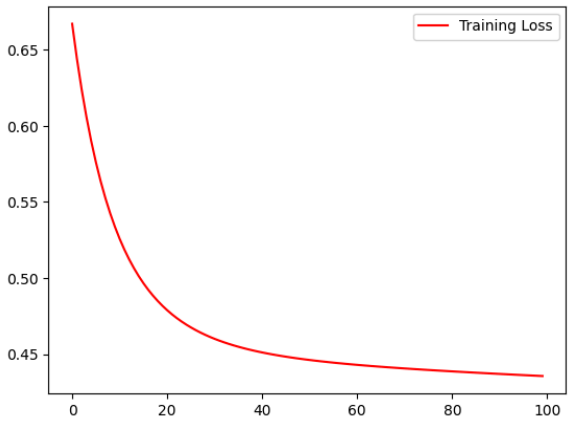
For both the training and validation loss, the graph consistently decreases as the number of epochs increases. It means that the model is learning effectively.

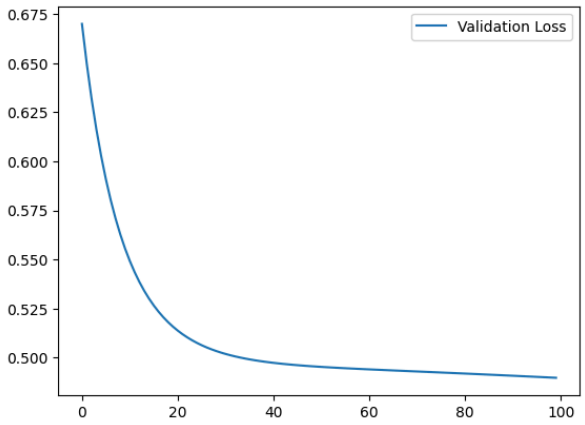
Both training and validation accuracy show increasing graphs as the number of epochs increases which means that the model is learning well on their training data. There are fluctuations in the graph in the validation accuracy.

Overall the model shows good convergence as the loss falls gradually as well as the accuracies increase gradually. Since the accuracies in both the graphs stagnate around same value (0.85 and 0.82) it means that there is not overfitting.

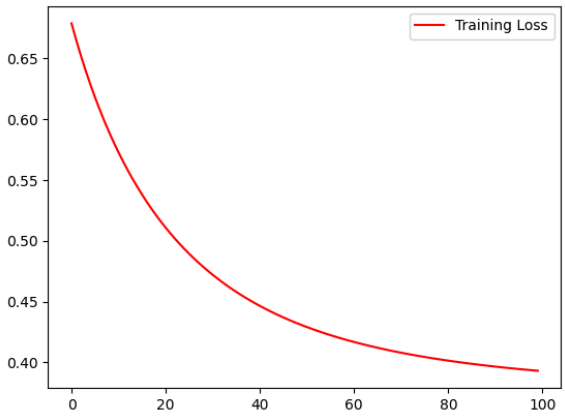
(b)

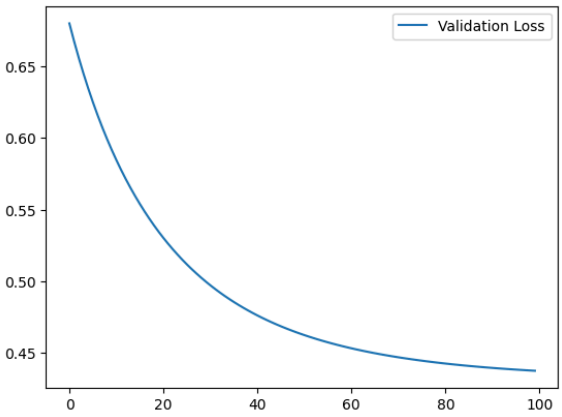
Min-Max Scaling:





No Scaling:





When Min-Max scaling is applied, the model converges early, showing low values of loss early, stagnating around epoch 40, whereas in no scaling, the model takes 100 epoch to converge. Min-max scaled loss is steeper than no scaling.

(c)
Confusion matrix:

```
----Without Scaling----
[[516   3]
 [110   7]]
----With Scaling----
[[519   0]
 [117   0]]
```

```
----Without Scaling----
Precision: 0.7
Recall: 0.05982905982905983
F1 Score: 0.11023622047244094
ROC-AUC Score: 0.5270243565041253

----After Scaling----
Precision: 0.0
Recall: 0.0
F1 Score: 0.0
ROC-AUC Score: 0.5
```

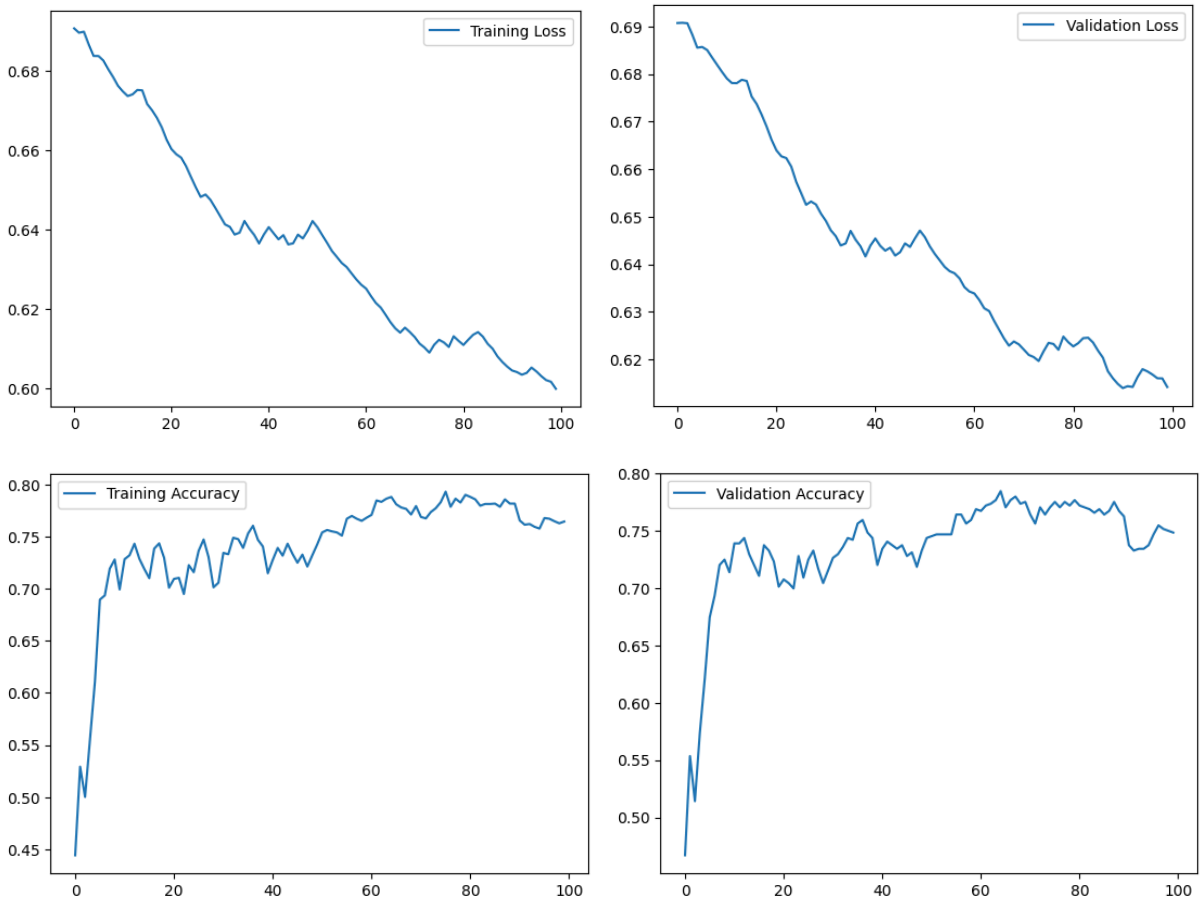
Without scaling:
TN: 516 FP: 3 FN: 110 TP: 7

After scaling:
TN: 519 FP: 0 FN: 117 TP: 0

Without scaling: The model is able to predict both the classes (0 and 1), even though true positives are quite low, all the scores are better than after scaled ones. Model is able to predict a class correctly 70% of the time. As we can see from the confusion matrix, the model is able to predict only some actual positive samples. F1 score is low because of low recall. ROC_AUC score being around 50% tells that the model is just better than the chance of getting a randomly guessed class right.
After scaling: The model is able to predict only negative class and no positive class. The precision is 0 and recall is also 0. Model is unable to predict any positive samples at all. The F1 score is also 0 because of this. a 50% ROC-AUC score is equivalent to the chance of guessing a class right by chance.

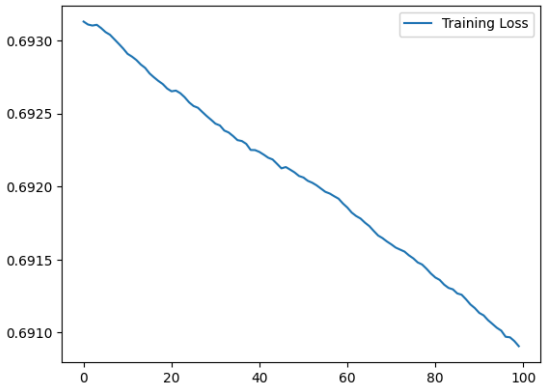
Scaling has affected the model negatively

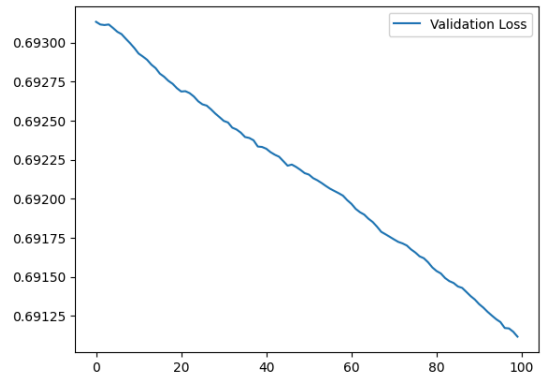
(d)
Stochastic Gradient Descent:

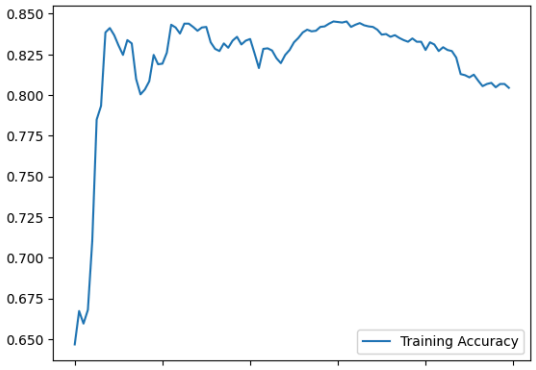


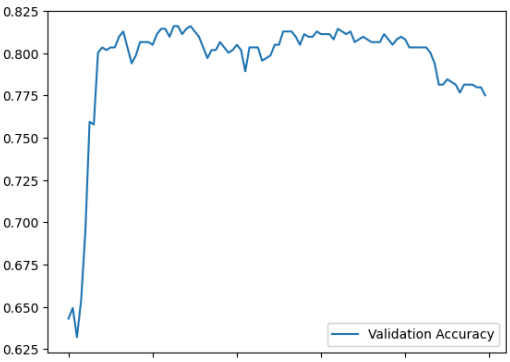
Stochastic Gradient Descent:
The model adjusts its weight on each epoch on randomly chosen features. Since average is not calculated before the weight is adjusted, the graph is bound to show unevenness and instability. Since because of random features, weights also may get randomized, accuracies can also show unevenness and instability. The graph will change if I run the code again, because a different feature is selected each time.

Mini-Batch Gradient Descent(Batch size: 4):

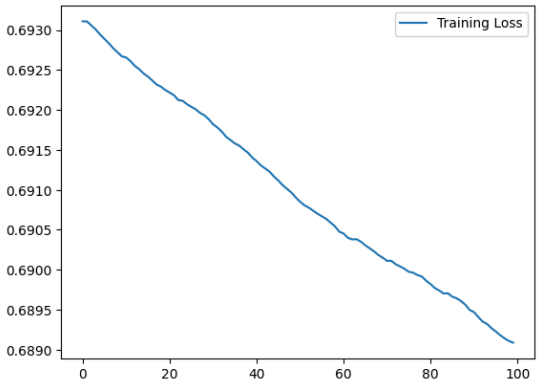


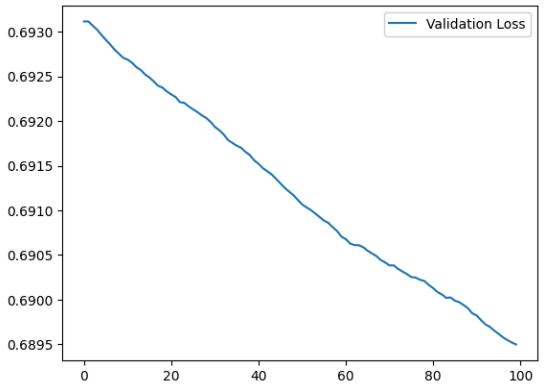


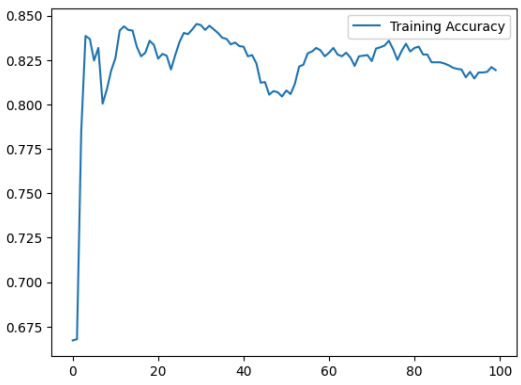


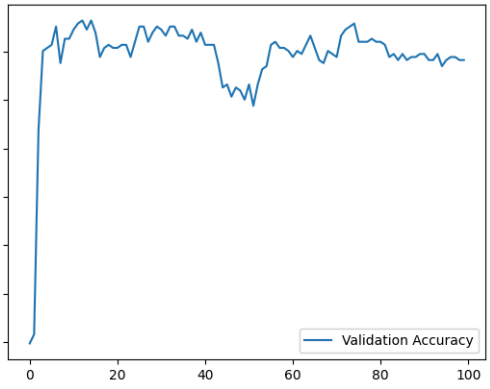


Mini-batch gradient descent(Batch size: 8):









Mini-Batch Gradient Descent:
This gradient descent updates its weights batch wise. Greater the batch size, greater there is unevenness and instability in the graph. A greater batch size is able to capture more variation in the data, leading to the given graphs.

Stochastic Gradient descent has more convergence speed, since it is based on randomness. It is overall faster and has a greater chance to reach MSE. On the other hand, mini-batch gradient descent can have slower convergence speed depending on the batch size. Greater the batch size, slower the speed.

Stochastic Gradient descent is more unstable as compared to Mini-Batch Gradient Descent.

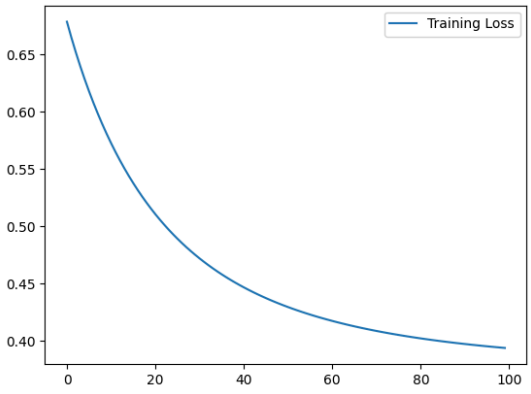
(e)

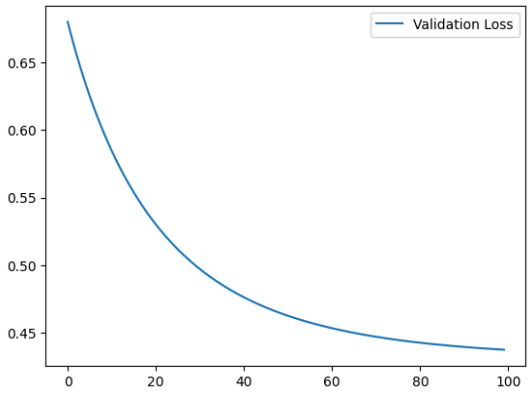
```
1 fold Accuracy: 0.8323494687131051, Precision: 0.5, Recall: 0.04225352112676056, F1 Score: 0.07792207792207793
2 fold Accuracy: 0.872491145218418, Precision: 0.9, Recall: 0.07758620689655173, F1 Score: 0.14285714285714285
3 fold Accuracy: 0.8476977567886659, Precision: 0.5384615384615384, Recall: 0.05384615384615385, F1 Score: 0.09790209790209789
4 fold Accuracy: 0.8547815820543093, Precision: 0.8, Recall: 0.06201550387596899, F1 Score: 0.11510791366906474
5 fold Accuracy: 0.8541176470588235, Precision: 0.5714285714285714, Recall: 0.09448818897637795, F1 Score: 0.16216216216216214
```

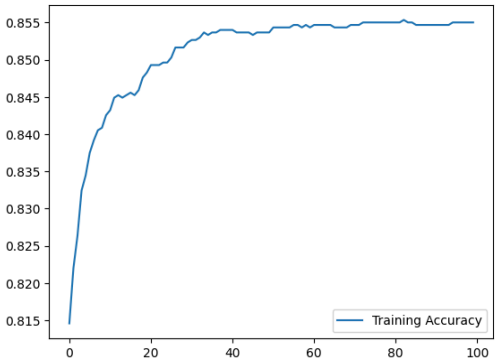
```
Average Accuracy: 0.8522875199666643
Average Precision: 0.6619780219780219
Average Recall: 0.066037914944362 62
Average F1 Score: 0.11919027890250913
Standard Deviation of Accuracy: 0.012933274898520 06
Standard Deviation of Precision: 0.158364726336133 38
Standard Deviation of Recall: 0.018293700326940315
Standard Deviation of F1 Score: 0.030264236216019 58
```

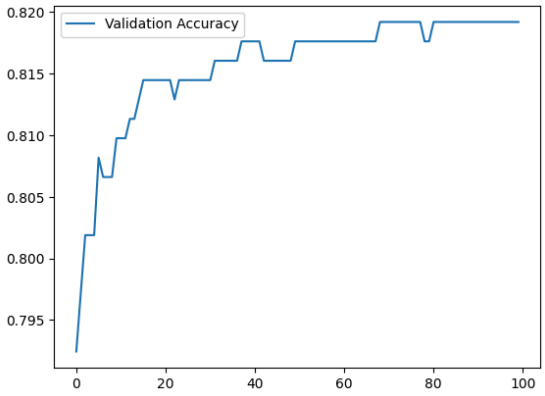
Accuracy is stable across the folds, ranging from 0.823 to 0.875. It is more or less consistent. This indicates low variance. It is stable.
Precision shows high variance. It ranges from 0.5 to 0.9. It is unstable.
Recall is low and unstable across the folds. It has high variance. It ranges from 0.0423 to 0.0945.
F1 score also has high variance. It is inconsistent and unstable. It ranges from 0.0779 to 0.1622.

(f) L1 regularization with early stopping:

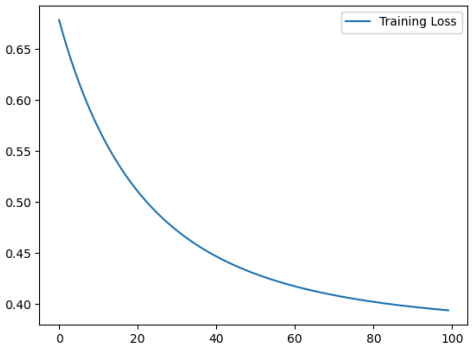


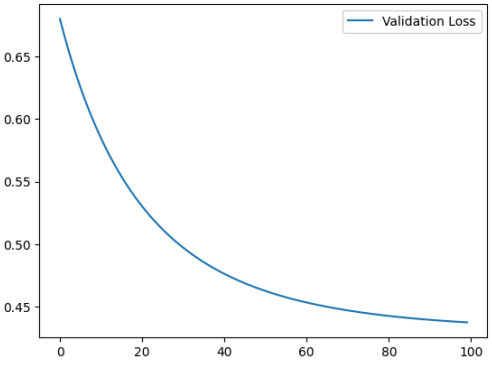


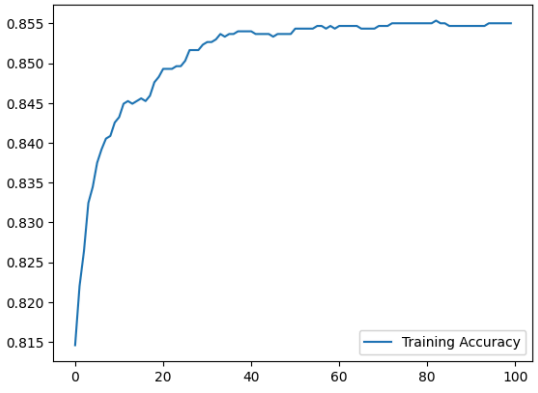


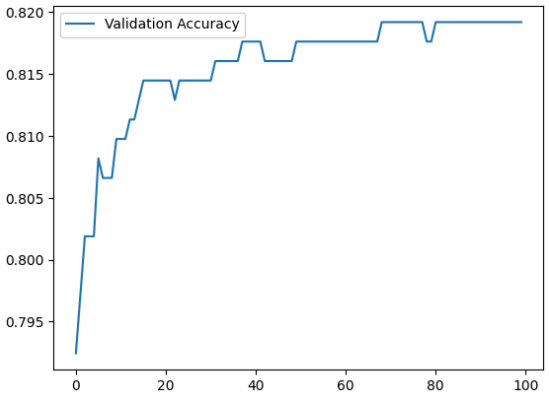


L2 regularization with early stopping:

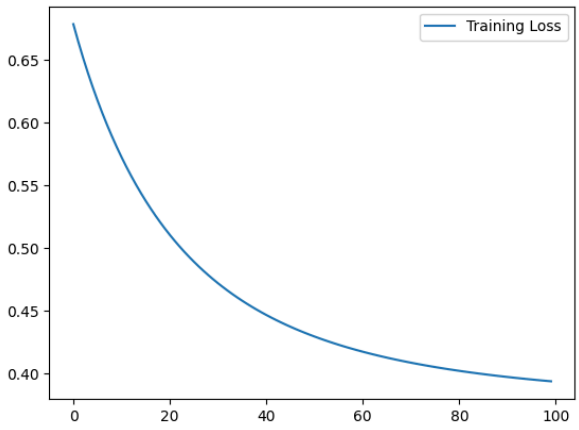


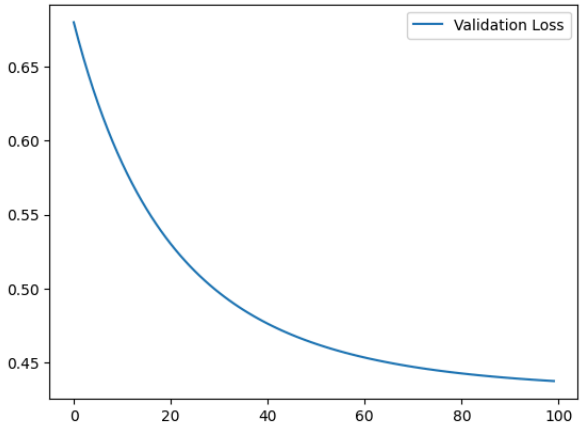


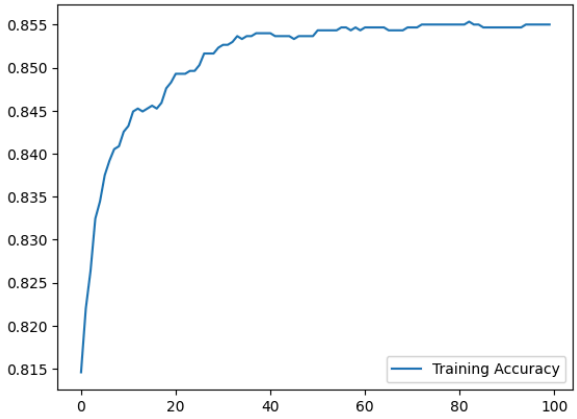


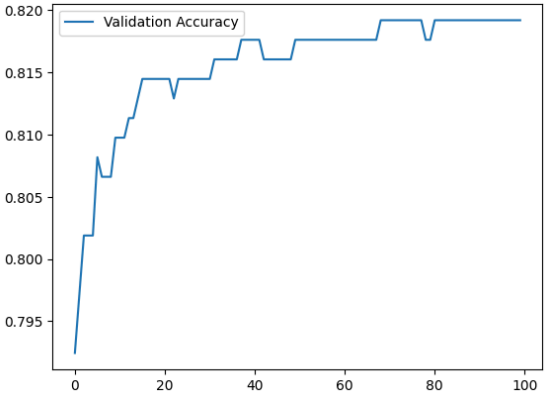


L1 regularization without early stopping

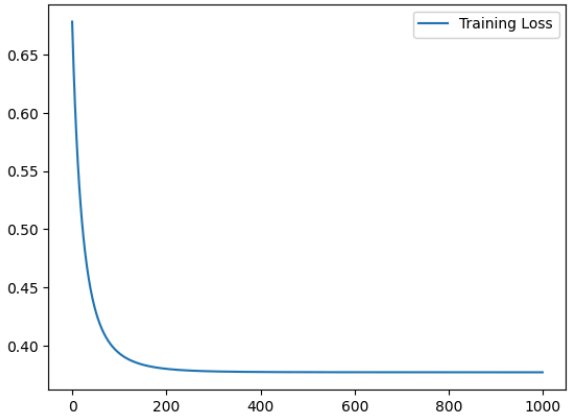


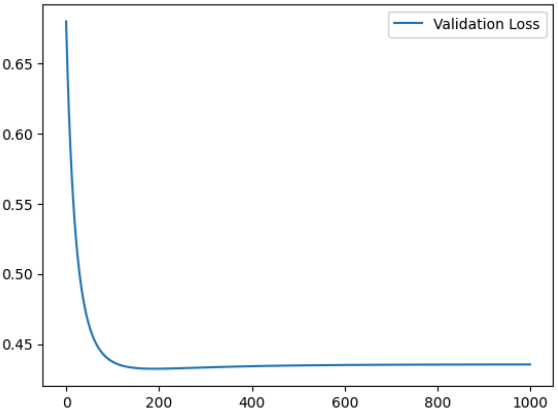


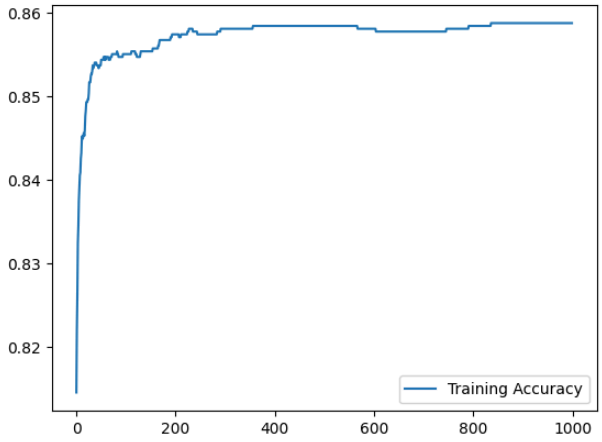


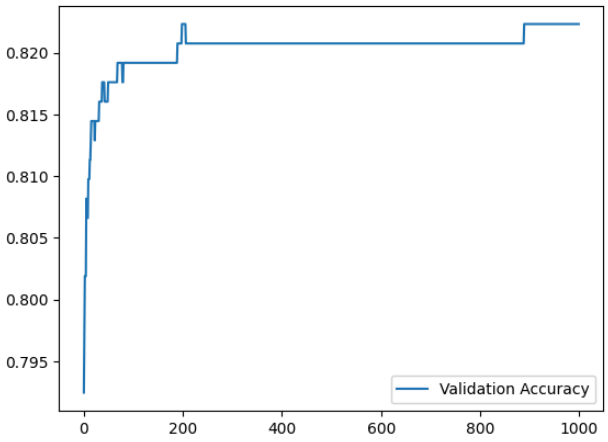


L2 regularization without stopping early







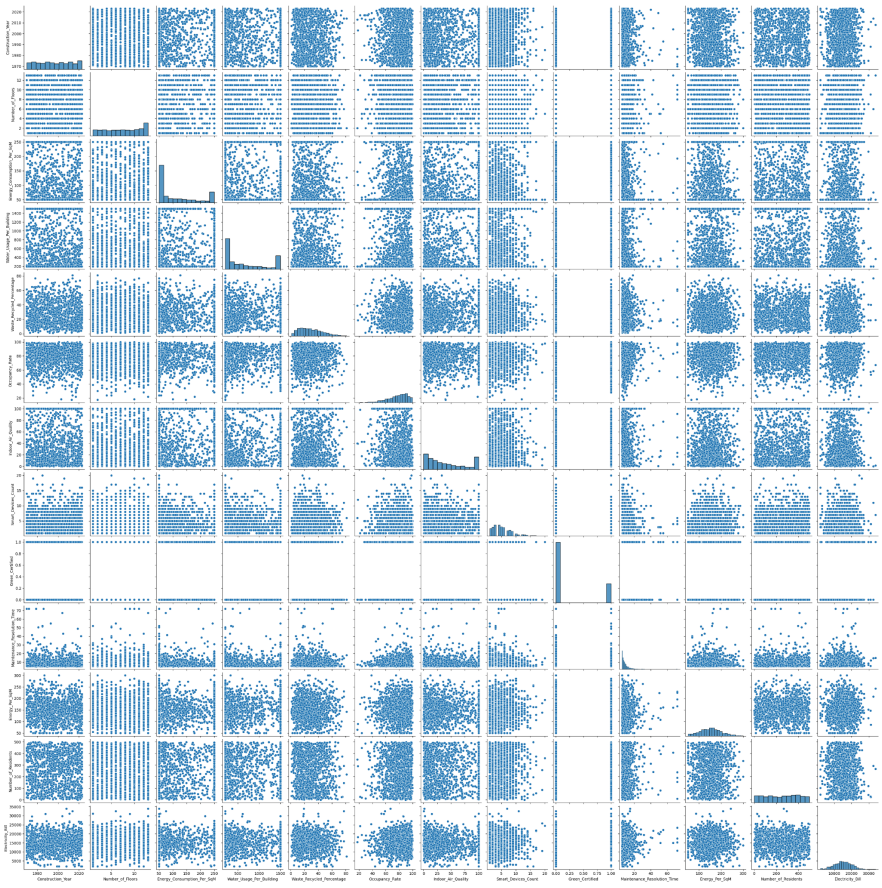


With early stopping we have prevented the code to run more since we have already reached highest accuracy.
Both L1 and L2 regularization show smooth training and validation loss. Though the accuracies are uneven due to the penalty term being subtracted from both the cases. There is a little bit of overfitting in L1 and L2 regularization with early stopping. On L2 regularization with stopping early there is also overfitting when the validation loss graph rises a little bit between epoch 200 and 800.

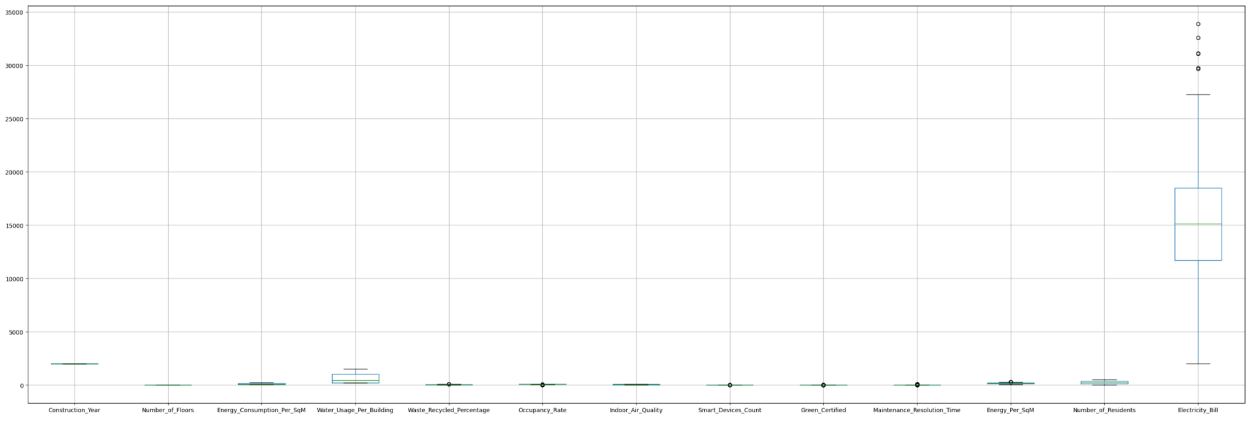
Section C:

(a) Plots:

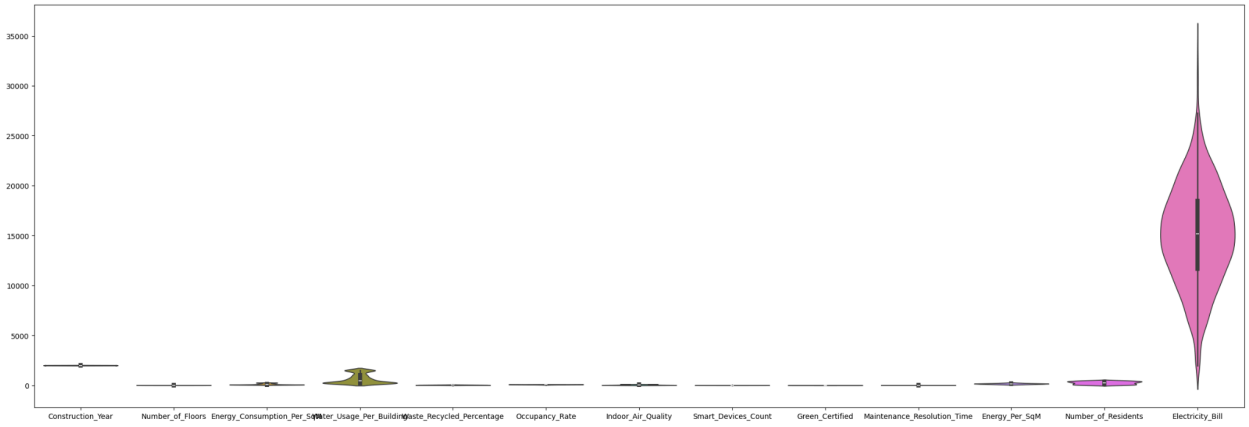
- Pair Plot:



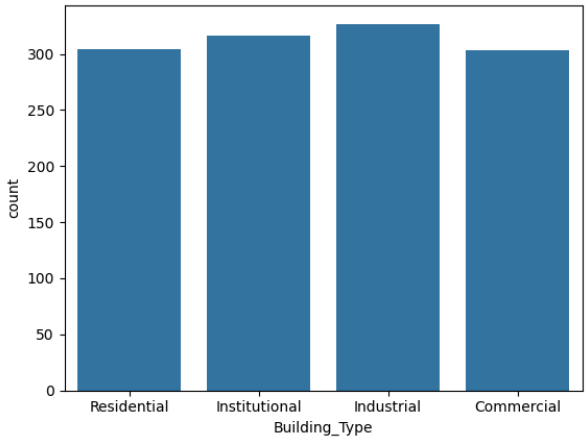
- Box Plot

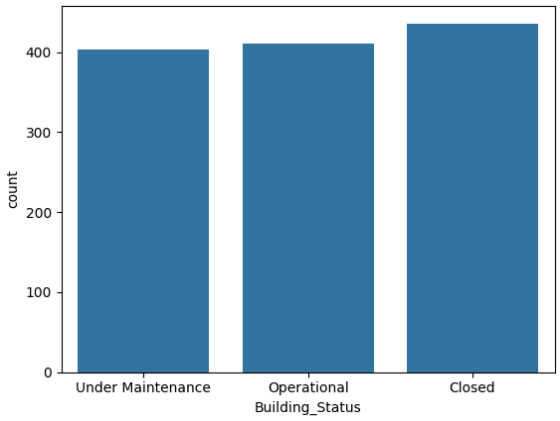


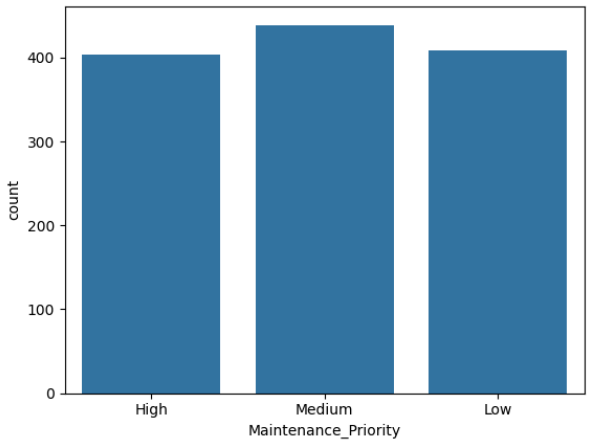
- Violin Plot

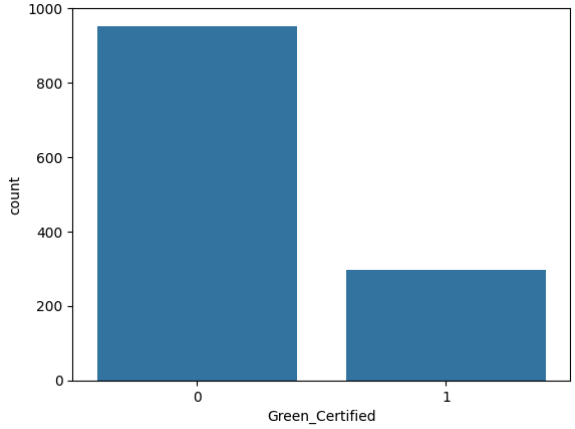


- Count Plots

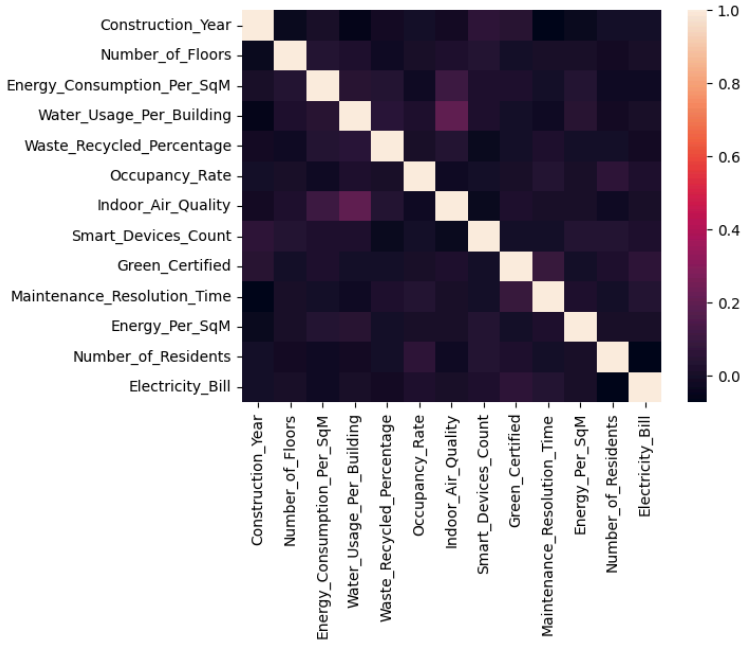








- Correlation Heatmap



Insights based on the visualizations are:

1. The box plot shows that there are outliers in the electricity bill column.
2. In the heatmap we can see that there is a relatively high correlation between the water_Usage_Per_Building column and Indoor_Air_Quality column.
3. In the box plot we can see that Electricity_Bill columns have more spread than the rest of the columns. It has larger values as compared to the other columns.
4. From the count plot of categorical features we can see that the count of green_certified as 1 home is far less , even less than half as that of green_certified 0.
5. In the heatmap we can see that there is a relatively high correlation between the Energy_Consumed_Per_SqM column and Indoor_Air_Quality column.

(b) We have got 2 separate clusters achieved after dimensionality reduction after UMAP is applied to the dataset.

(c) I have applied Label Encoding on non-numerical categorical features: Building_Type, Building_Status and Maintainence_Priority. For normalization, I have used Min-Max Scaling.

Scores in train and test data:

```
Scores on train data :
MSE:                24461325.591933943
RMSE:               4945.839220186393
R2 score:           0.018021482706563452
Adjusted R2 score: 0.003052297991724462
MAE:                3994.575124763379

Scores on test data :
MSE:                24388436.567185808
RMSE:               4938.465001109738
R2 score:           -0.026482604884292993
Adjusted R2 score: -0.09228277186405531
MAE:                3924.6655931301643
```

(d) After Perform Recursive Feature Elimination (RFE), we got features: Smart_Devices_Count, Maintenance_Resolution_Time and Number_of_Residents. After training the regression model on these features we get the following results:

```
Scores on train data after RFE :
MSE:                 24727502.02294619
RMSE:                4972.675539681449
R2 score:            0.0073606627328374894
Adjusted R2 score: -0.007796009481600974
MAE:                 3993.0307790454412

Scores on test data after RFE :
MSE:                 24036672.468793906
RMSE:                4902.720925036821
R2 score:            -0.011677238946735669
Adjusted R2 score: -0.07652834400742381
MAE:                 3871.500032034888
```

Observations:
R2 score tells us how well our regression line fits the data.
Train data: There is an increase in both MSE and RMSE scores after RFE has been used. Model performance has decreased slightly. MAE has dropped slightly indicating that the model is showing small improvements. R2 score has dropped slightly but adjusted R2 score has dropped significantly indicating a reduced fit.

Test data: There is a drop in the MSE and RMSE scores which means the model's performance has improved on test data after RFE. R2 and adjusted R2 scores have increased slightly but still the performance is poor. MAE has decreased showing minor improvement in model's accuracy.

(e)

Scores after ridge regression:

```
Scores on train data (Ridge Regression):
MSE:                24232846.06543975
RMSE:               4922.68687460819
R2 score:           0.027193593425397844
Adjusted R2 score: 0.012364227471516664
MAE:                3965.903551051294

Scores on test data (Ridge Regression):
MSE:                23893973.745839164
RMSE:               4888.146248409428
R2 score:           -0.005671205864265394
Adjusted R2 score: -0.07013730880428248
MAE:                3871.022155527163
```

Scores of part(c):

```
Scores on train data :
MSE:                 24461325.591933943
RMSE:                4945.839220186393
R2 score:            0.018021482706563452
Adjusted R2 score: 0.003052297991724462
MAE:                 3994.575124763379

Scores on test data :
MSE:                 24388436.567185808
RMSE:                4938.465001109738
R2 score:            -0.026482604884292993
Adjusted R2 score: -0.09228277186405531
MAE:                 3924.6655931301643
```

Observations:
Train data: There is a slight drop in MSE and RMSE scores which means there is an increase in model's performance on the training set. R2 score and adjusted R2 score has also increased slightly meaning the ridge regression has led to a better fit. MAE has dropped indicating a better performance.
Testing data: Both MSE and RMSE have dropped indicating better performance on the testing set. Both R2 score and adjusted R2 score have increased slightly even though they are still negative. Ridge regression has increased model performance.

(f)

```
Scores on train data with 4 components :
Scores on train data :
MSE:                24840421.173138566
RMSE:               4984.016570311395
R2 score:           0.002803022229239227
Adjusted R2 score: -0.012398151212388298
MAE:                3995.726626026647

Scores on test data :
MSE:                24063463.71200572
RMSE:               4905.452447227036
R2 score:           -0.01280485305370771
Adjusted R2 score: -0.07772824106997112
MAE:                3892.936737095693
```

```
Scores on train data with 5 components :
Scores on train data :
MSE:                24840421.17072989
RMSE:               4984.016570069755
R2 score:           0.0028030223259334353
Adjusted R2 score: -0.012398151114219935
MAE:                3995.728357640094

Scores on test data :
MSE:                24063465.17021065
RMSE:               4905.452595858067
R2 score:           -0.01280491442795717
Adjusted R2 score: -0.07772830637846728
MAE:                3892.937772377163
```

```
Scores on train data with 6 components :
Scores on train data :
MSE:                24652036.676067207
RMSE:               4965.081739112379
R2 score:           0.01036555306674214
Adjusted R2 score: -0.0047203378926061745
MAE:                3962.174500170394

Scores on test data :
MSE:                23774826.62332753
RMSE:               4875.943664905034
R2 score:           -0.0006564338700341121
Adjusted R2 score: -0.06480107706683125
MAE:                3860.9561120841954
```

```
Scores on train data with 8 components :
Scores on train data :
MSE:                24528000.44149878
RMSE:               4952.575132342646
R2 score:           0.015344879197470296
Adjusted R2 score: 0.00033489259987085074
MAE:                3966.140619735859

Scores on test data :
MSE:                23624642.63404084
RMSE:               4860.518761823767
R2 score:           0.005664645880618657
Adjusted R2 score: -0.05807479989626474
MAE:                3863.6110990018433
```

MSE: It decreases as the number of components increases. This means more components are able to capture more relationships between features and labels during training.

RMSE: It also follows a similar decreasing order as MSE.

R2 Score: Slight improvement as components increase, highest when components are 8. Still it is underfitting.

Adjusted R2 score: Follows the same trend as R2 score.

MAE: Decreases as components increase. Make the model's performance better.

Therefore, an increase in components can lead to better performance, even though in this case it is not that significant.

(g)

```
-----------Scores on test data----------
Scores on test data for alpha=0.1 :
MSE:                24124832.210987333
RMSE:               4911.70359559566
R2 score:           -0.015387786015357285
Adjusted R2 score: -0.08047674665736726
MAE:                3904.822286165877

Scores on test data for alpha=0.5 :
MSE:                24026419.025244173
RMSE:               4901.675124408407
R2 score:           -0.01124568272889559
Adjusted R2 score: -0.07606912392946574
MAE:                3891.1400571850363

Scores on test data for alpha=1 :
MSE:                24022773.951679796
RMSE:               4901.30329113388
R2 score:           -0.011092265571668314
Adjusted R2 score: -0.07590587233908286
MAE:                3888.845512310233
```

```
Scores on test data for alpha=2 :
MSE:                24029032.414446905
RMSE:               4901.9416983932915
R2 score:           -0.01135567742039667
Adjusted R2 score: -0.07618616956272972
MAE:                3887.8693996321213

Scores on test data for alpha=5 :
MSE:                24038052.31124536
RMSE:               4902.8616451257685
R2 score:           -0.011735315005444669
Adjusted R2 score: -0.076590142890409
MAE:                3887.268110652529

Scores on test data for alpha=10 :
MSE:                24042375.92768497
RMSE:               4903.302553145682
R2 score:           -0.011917291289706533
Adjusted R2 score: -0.07678378432109811
MAE:                3887.113965973759
```

```
Scores on test data for alpha=20 :
MSE:                24044870.770278953
RMSE:               4903.556950855058
R2 score:           -0.012022296484180295
Adjusted R2 score: -0.07689552061778149
MAE:                3887.0384421264444

Scores on test data for alpha=50 :
MSE:                24046489.041075364
RMSE:               4903.721957969819
R2 score:           -0.012090407730997343
Adjusted R2 score: -0.07696799797016385
MAE:                3887.0085197029953

Scores on test data for alpha=100 :
MSE:                24047234.200455356
RMSE:               4903.797936340297
R2 score:           -0.012121770673813259
Adjusted R2 score: -0.07700137135803198
MAE:                3887.025310728601
```

Alpha	MSE	RMSE	R2 score	Adjusted R2	MAE
0.1	24124832.21	4911.7	-0.0154	-0.0805	3904.82
0.5	24026419.03	4901.68	-0.0112	-0.0761	3891.14
1.0	24022773.95	4901.3	-0.0111	-0.0759	3888.85
2.0	24029032.41	4901.94	-0.0114	-0.0762	3887.87
5.0	24038052.31	4902.86	-0.0117	-0.0766	3887.27
10.0	24042375.93	4903.3	-0.0119	-0.0768	3887.11
20.0	24044870.77	4903.56	-0.012	-0.0769	3887.04
50.0	24046489.04	4903.72	-0.0121	-0.077	3887.01
100.0	24047234.2	4903.8	-0.0121	-0.077	3887.03

MSE and RMSE: Slight decrease in when alpha goes from 0.1 to 1 but after that it increases.
R2 score: It is negative throughout all the alphas.
Adjusted R2 score: follows a similar trend as R2 score. It indicates poor model performance.
MAE: It decreases from 0.1 to 5 after it remains mostly the same.

(h)

```
--------------------Scores on test data (Gradient Boosting Regressor)------------------
MSE:                27727174.345418327
RMSE:               5265.659915472925
R2 score:           -0.16700642412069677
Adjusted R2 score: -0.2418145282309978
MAE:                4160.909631457792
```

As compared to (c) MSE is higher, RMSE is also higher, R2 score is extremely low, a poor fit. Adjusted R2 score is negative similar to R2 score. MAE is higher too. Overall the model performs poorly than that in (c) .

As compared to (g) MSE here is still higher than those across all alpha. RMSE is also significantly high. R2, though negative in both parts, is still better in (g). Same is the case with adjusted R2 score. MAE is also high here among all alphas in (g).