REPORT

Name - Himang Chandra Garg

Assignment -1

Roll no. - 2022214

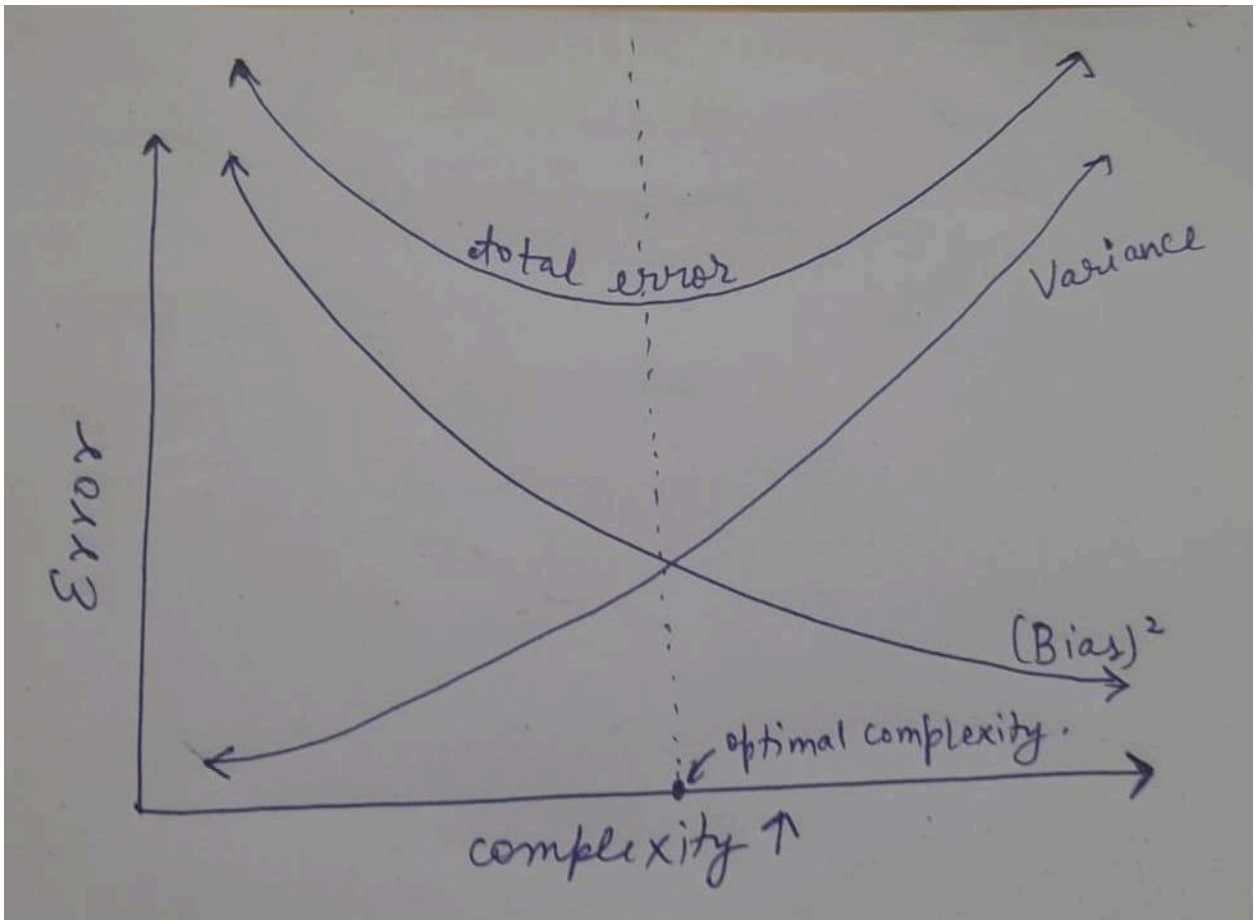
SECTION A

(a). **Overfitting** occurs by adding more features or by including higher-order polynomial terms in a regression model. As we make the model complex, bias decreases, and variance increases.

Bias: In simple words, it measures the difference(error) between the predicted value and the expected value. Complex models generally have a low bias as they predict the value near the real value of the data as a whole.

Variance: It measures the deviation of the predicted values from the central value. In simple terms, it is the spread of the data. Complex models tend to fit the training data more closely hence the model is overly trained on the training data. When test data comes, it tends to give vague results which results in high variance.

We can see this phenomenon using this graph:


total error
Variance
Error
(Bias)²
optimal complexity.
complexity ↑

(b).

(b) True positive - 200
False positive - 20
True negative - 730
False negative - 50

* confusion matrix:

200	50
20	730

$$* \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{200+730}{1000} = 0.930$$

$$* \text{Precision} = \frac{TP}{TP+FP} = \frac{200}{220} = 0.909$$

$$* \text{Recall} = \frac{TP}{TP+FN} = \frac{200}{250} = 0.8$$

$$* \text{F1 Score} = \frac{2 \times \text{precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times 0.909 \times 0.8}{0.909+0.8}$$

$$= \frac{1.45}{1.709} = 0.848.$$

These performance metrics suggest that the performance of the model is reasonably good as it achieved 93% accuracy overall. High precision suggests that It lets very few legitimate emails be identified as spam emails. Recall is somewhat low, which means a relatively higher number of spam emails have slipped into the inbox. The F1 score of 0.848 which is the harmonic mean of precision and recall also suggests that the model has performed well.

(c).

(c) given data:

x	y	x^2	xy
3	15	9	45
6	30	36	180
10	55	100	550
15	85	225	[redacted] 1275
18	100	324	[redacted] 1800

$\Sigma x = 52$

$\Sigma y = 285$

$\Sigma x^2 = 694$

$\Sigma xy = 3850$

$\bar{x} = 52/5 = 10.4$, $\overline{x^2} = 694/5 = 138.8$

$\bar{y} = 285/5 = 57$, $\overline{xy} = 3850/5 = 770$

let, the equation the regression line, $y = mx + c$.

where, $m = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2} = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - (\bar{x})^2}$

$= \frac{770 - 10.4 \times 57}{138.8 - (10.4)^2}$

$\boxed{m = 5.783}$

$c = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2} = \frac{\bar{y}\,\overline{x^2} - \bar{x}\,\overline{xy}}{\overline{x^2} - (\bar{x})^2}$

$= \frac{57 \times 138.8 - 10.4 \times 770}{138.8 - (10.4)^2}$

$= \frac{7911.6 - 8008}{30.64}$

$\boxed{c = -3.146}$

regression line: $\boxed{y = (5.783)x - 3.146}$

prediction of $x = 12$:

$y = 5.783 \times 12 - 3.146$

$y = 69.369 - 3.146$

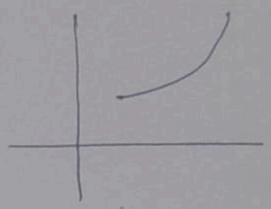
$\boxed{y = 66.2[redacted]5}$.

(d).

(d) Empirical Risk is the ~~risk~~ purely based upon training data.

let, the training data be;

X	y
1	1
2	4
3	9



let, f1 & f2 be two models, f1 being a complex one f1 being simple one.

Clet the loss be $|y-\hat{y}|$

f1 predictions after training: —

Expirical Risk = 0

X	f1(x)
1	1
2	4
3	9

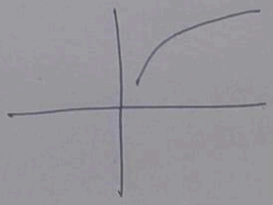
f2 predictions after training: —

Expirical Risk = $\frac{1+1+1}{3} = 1$

X	f2(x)
1	2
2	5
3	8

let, the testing data be,

X	y
1	1
2	6
3	9



~~Expirical~~ Test loss for f1 = 4/3 = 1.33

Test loss for f2 = 3/3 = 1

⇒ Hence, we see that the model with higher ~~risk~~ empirical Risk performed better in this case because f1 does not generalise while f2 ~~overfitting~~ generalised better hence high empirical risk.

SECTION B

Preprocessing the data set -

1. Filling missing categorical values with mode.
2. Filling missing numerical values with mean.
3. Since the data is very biased towards no heart disease, we have to do undersampling to make the balance.
4. Split the data set into train, test, and validation.

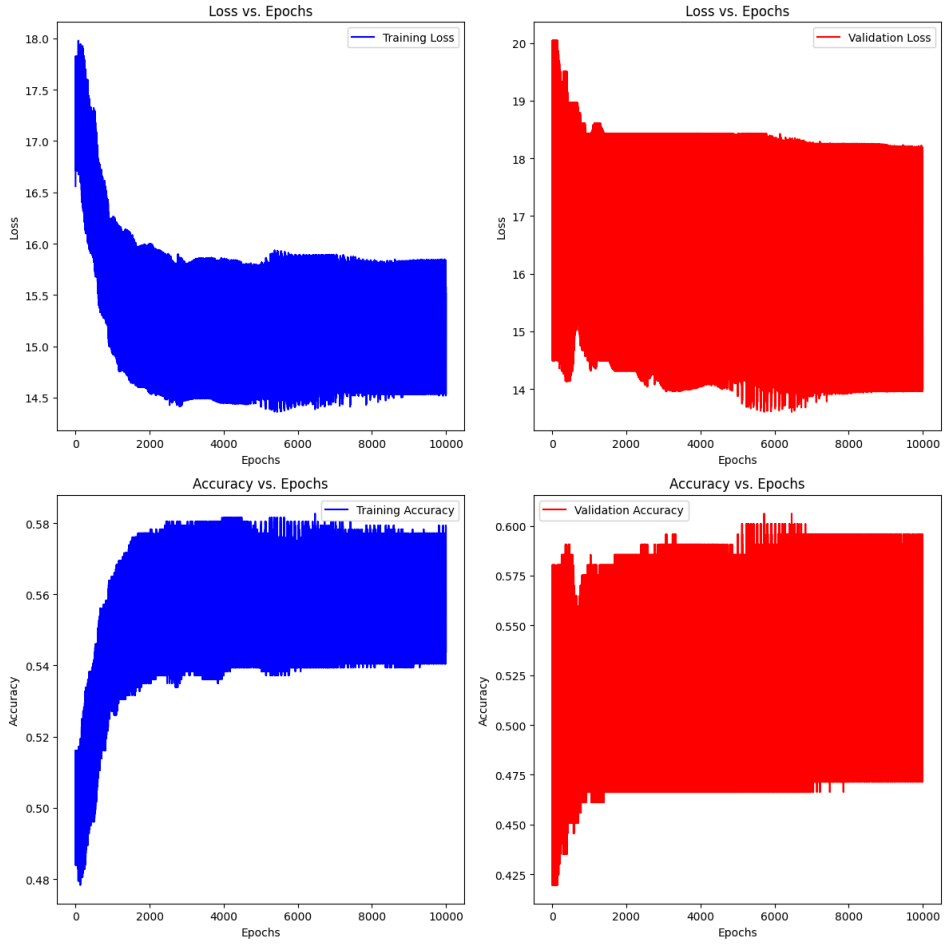
(a).

I have written a function for Logistic Regression using Batch Gradient Descent from scratch.

Parameters - train data, val data, learning rate, number of epochs.

Results - final updated weights and bias, loss, and accuracy history on training data and validation data, final values of y predicted after all the epochs.

These plots are without any kind of scaling:
Learning rate = 0.1 and number of epochs = 10000



The model did not converge well because the data entered was not normalized (scaled), but it still converged to a certain level of accuracy after around 1000 epochs.

The model kind of overfits the data because the training loss continued to decrease after 1000 epochs but the val loss almost remained stagnant.

The data is not normalised hence the weights and bias switching very fast and thus the graph became zigzag, thus it is looking like colour is filled.

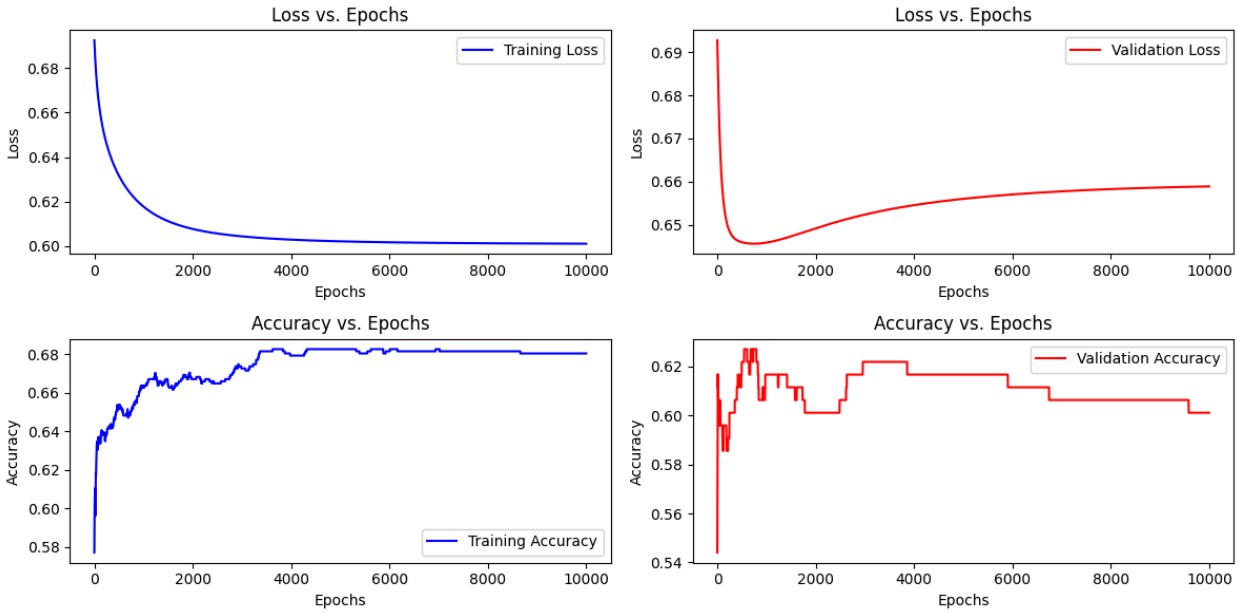
(b).

I implemented a function to do min-max scaling without using any library and again used the function written in part (a) to plot the graphs -

Parameters - train data, val data, learning rate, number of epochs.

Results - final updated weights and bias, loss, and accuracy history on training data and validation data, final values of y predicted after all the epochs.

Learning rate = 0.1 and number of epochs = 10000
Min-max scaling applied



This time the graph converges very well as shown. This also shows overfitting clearly after around 700 epochs the validation loss started to rise. Hence, scaling enhances the performance of the model.

(c).

I used the final prediction value result from the function I wrote in part (a) to find the confusion matrix, precision, Recall, F1 score, and ROC-AUC score using the sklearn library.

```
Confusion Matrix:
[[57 24]
 [53 59]]
Precision: 0.7108
Recall: 0.5268
F1 Score: 0.6051
ROC-AUC Score: 0.6152
```

These scores tell about the performance of the model. Precision measures the proportion of true positive predictions out of all positive predictions made by the model. Recall measures the proportion of true positives that the model identified. The F1 score is the harmonic mean of precision and recall. The ROC-AUC score represents the model's ability to distinguish between classes.

(d).

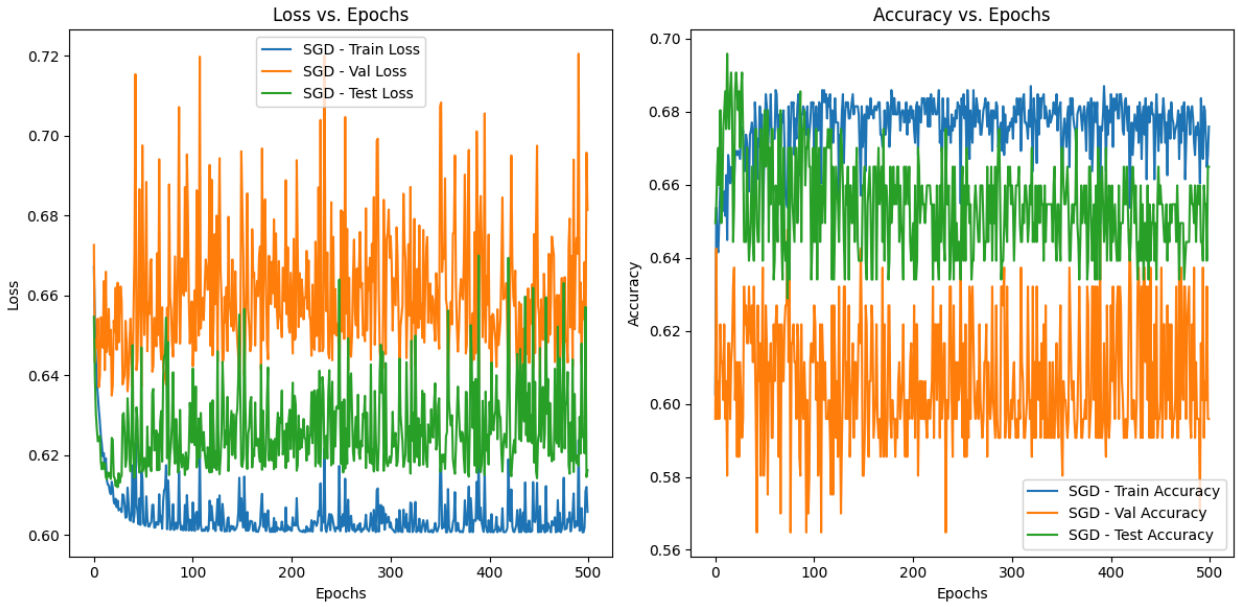
1. **Stochastic Gradient Descent**-

I added an additional for loop in the function written in part(a) that updates weight and bias on every iteration. Rest everything is same.

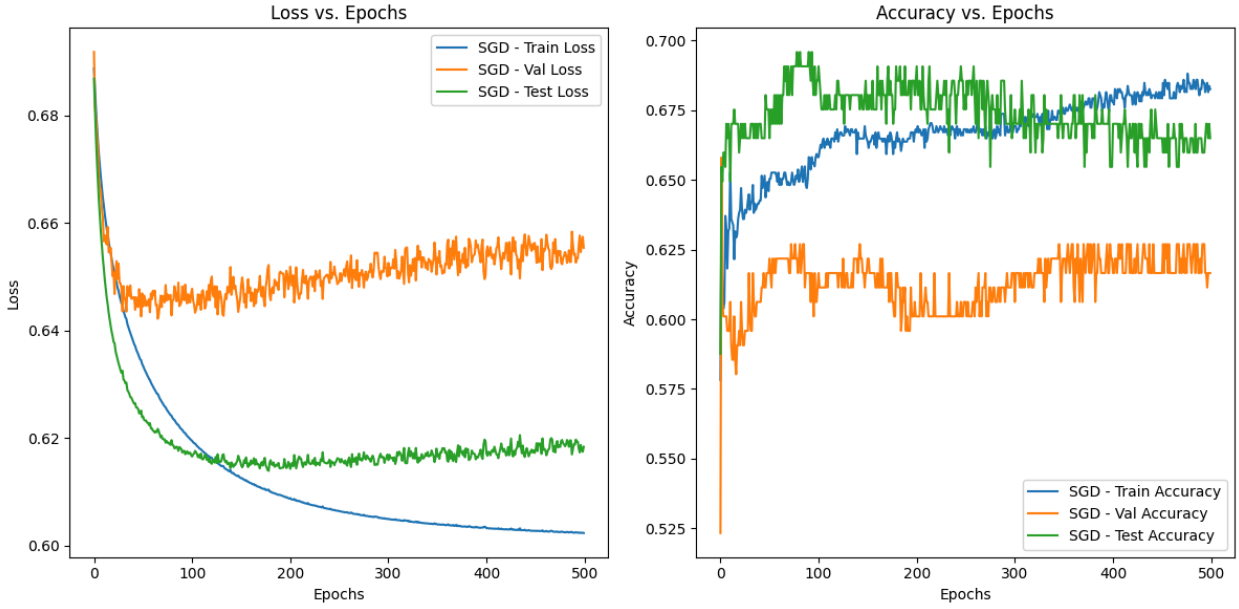
Parameters - train data, val data, test data, learning rate, number of epochs.

Results - final updated weights and bias, loss, and accuracy history on training data, validation data, test data, and final values of y predicted after all the epochs.

Learning rate = 0.1 and number of epochs = 500 (fewer epochs due to more computation time due to extra for loop)
Min-max scaling applied



Learning rate = 0.001(decreased learning rate because values are varying too much) and number of epochs = 500
Min-max scaling applied



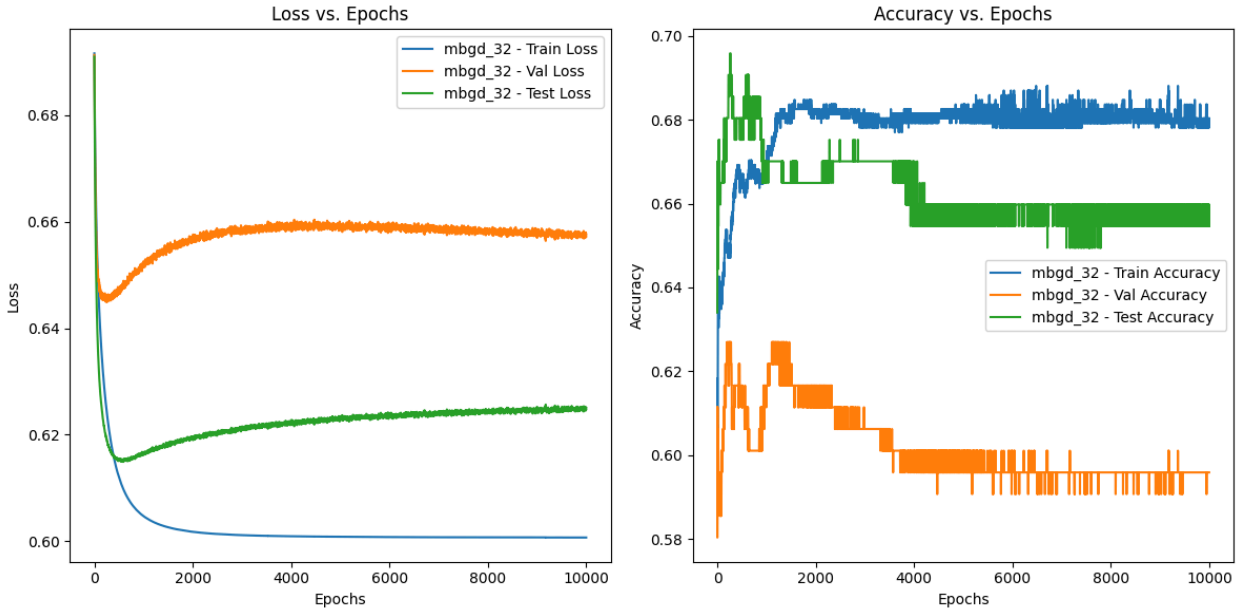
2. **Mini-Batch Gradient Descent**-

I added an additional for loop in the function written in part(a) that updates weight and bias on every batch iteration. Rest everything is same.

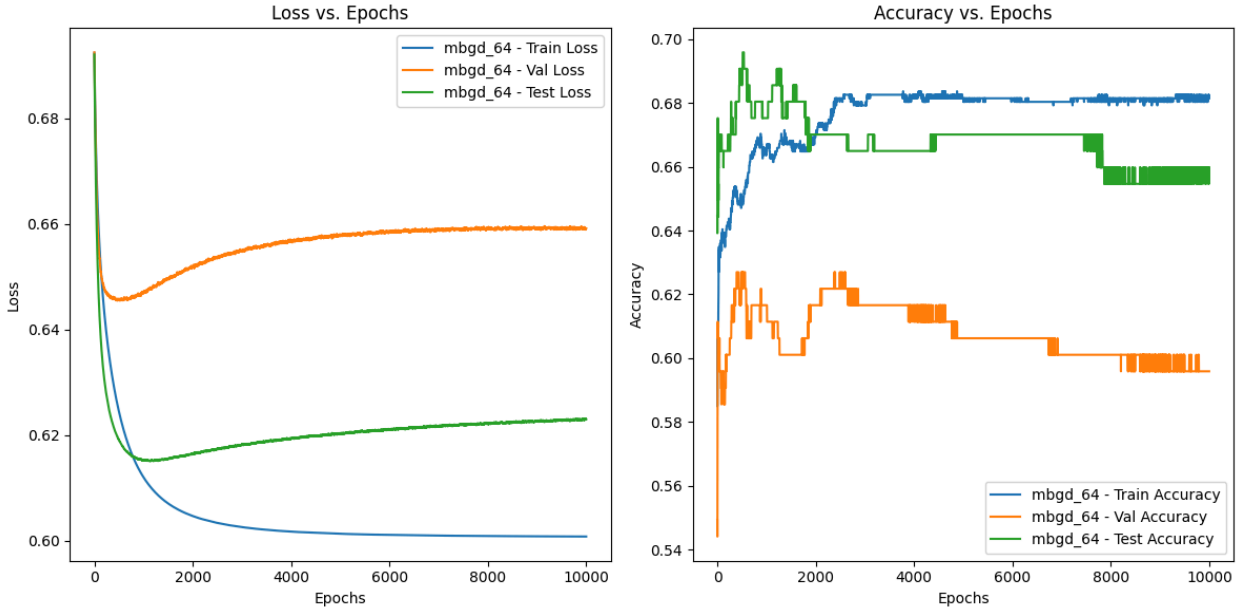
Parameters - train data, val data, test data, learning rate, number of epochs.

Results - final updated weights and bias, loss, and accuracy history on training data, validation data, test data, and final values of y predicted after all the epochs.

Learning rate = 0.01 and number of epochs = 10000 Batch Size = 32
Min-max scaling applied



Learning rate = 0.01 and number of epochs = 10000 Batch Size = 64
Min-max scaling applied



Tradeoffs:

1. **Convergence Speed:**

   SDG- very fast convergence initially but due to high variance not very optimal
   MBDG with 32- faster convergence
   MBDG with 64- slower convergence

2. **Stability:**

   SDG- very unstable and noisy (requiring reduced learning rate)
   MBDG with 32- stable and a little noisy
   MBDG with 64- more stable and negligible noise.

(e).

Implemented a function to do K-fold cross-validation. Split the data set into 5 parts. Ran a for loop from 1 to 5 and at each iteration used the Logistic Regression Batch gradient GD function of part (a). The weights and bias returned from the function were used to calculate accuracy, precision, recall, and F1 score.

Learning rate =0.1 and epochs = 1000

```
Accuracy: Mean = 0.5447, Std = 0.0210
Precision: Mean = 0.8277, Std = 0.0708
Recall: Mean = 0.1106, Std = 0.0177
F1 Score: Mean = 0.1945, Std = 0.0280
```

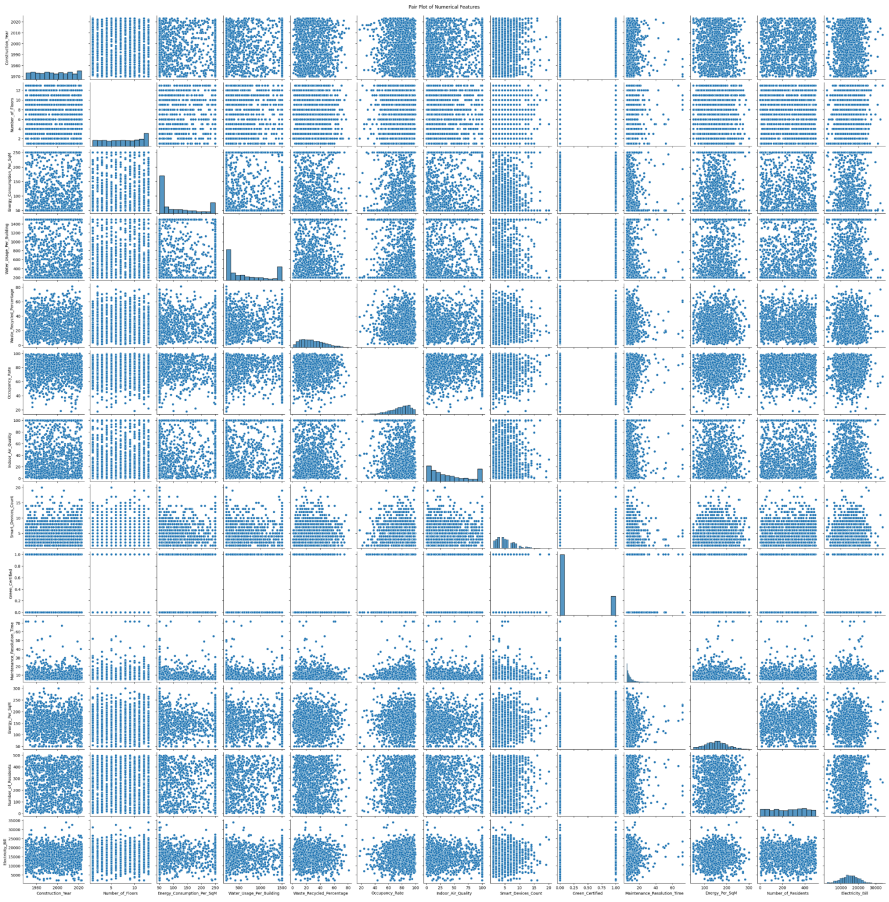
The model performed well across all the five folds. Stability is also high, and variance is good.

SECTION C

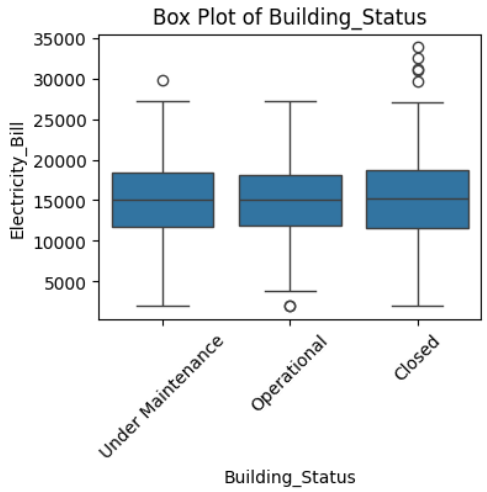
Split the data set into 80:20 using sklearn train-test split.

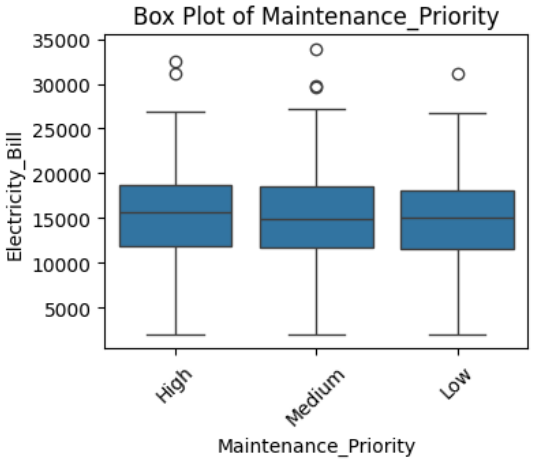
(a). Performed EDA on the complete data set using library functions of sklearn.

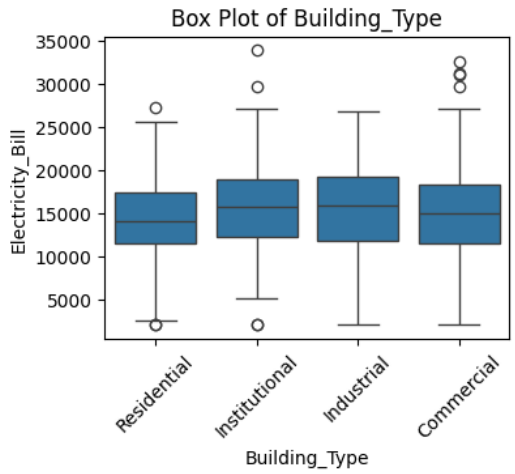
Pair plots-



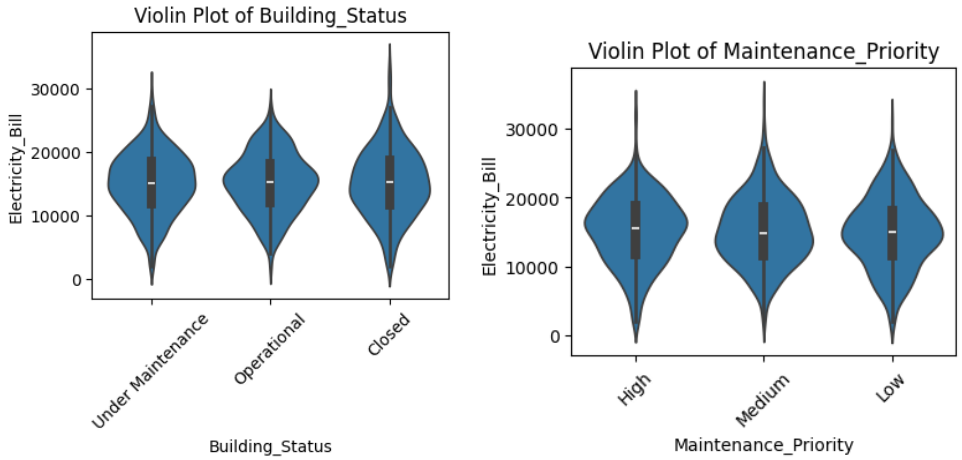
Box plots-

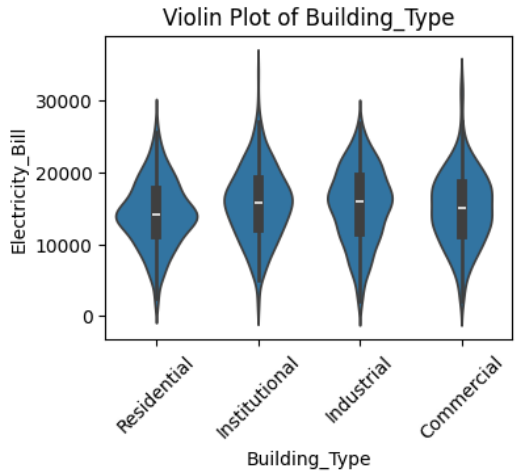

Box Plot of Building_Status
Electricity_Bill
35000
30000
25000
20000
15000
10000
5000
Under Maintenance
Operational
Closed
Building_Status


Box Plot of Maintenance_Priority
Electricity_Bill
35000
30000
25000
20000
15000
10000
5000
High
Medium
Low
Maintenance_Priority

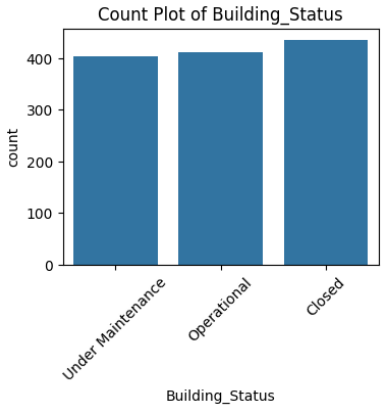


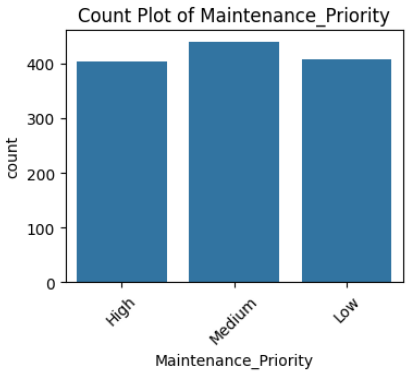
Violin plots:

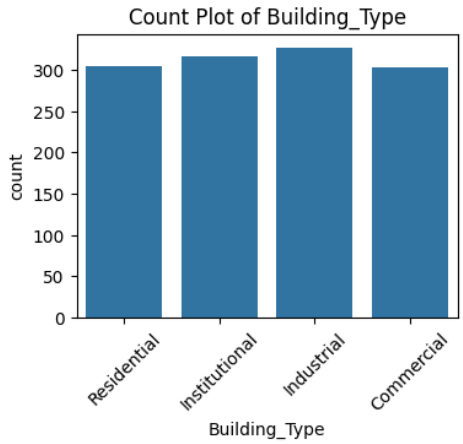




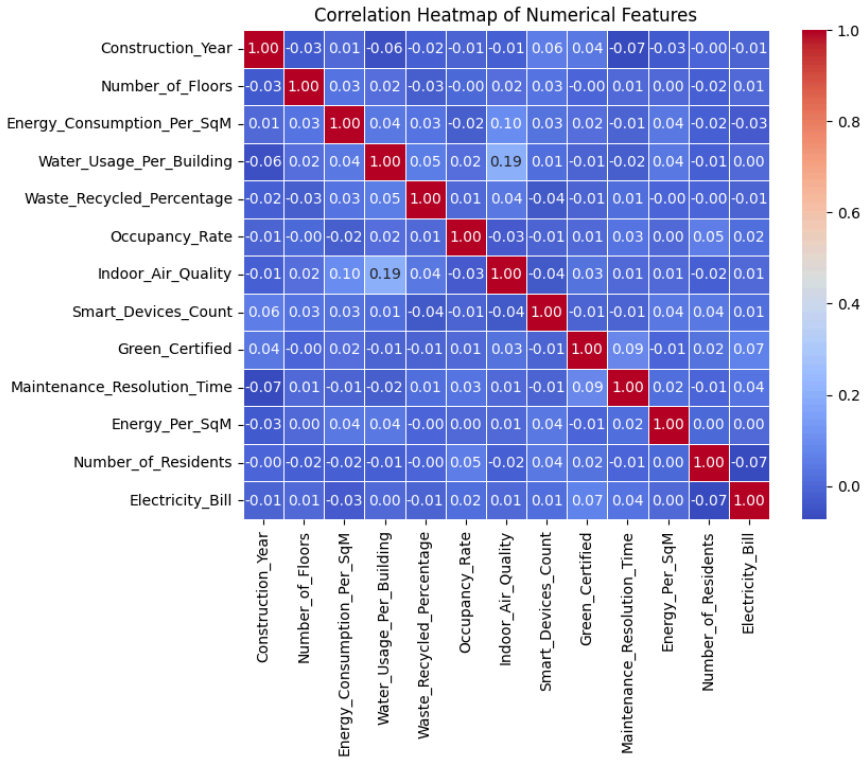
Count plot:







Correlation Heatmap:

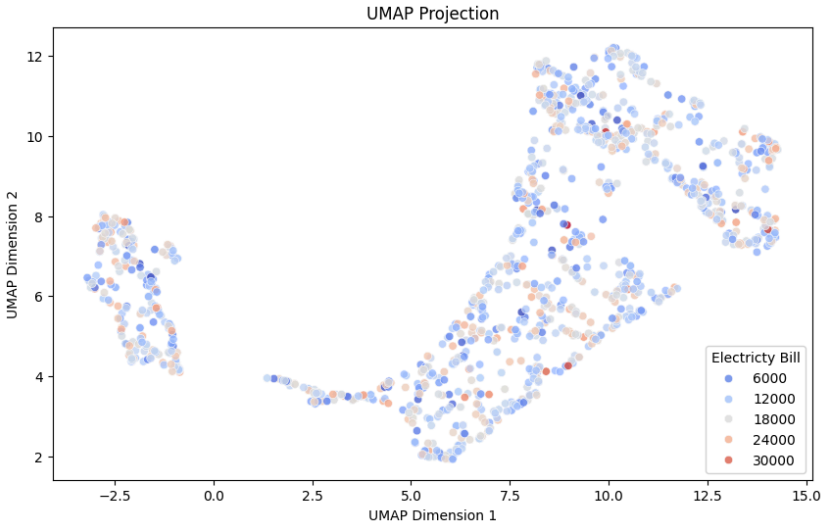


Insights about this data-

1. Residential Builidings have less electricity bill in comparison to other building types while industrial buildings have high.
2. Indoor air quality is most related to water usage per building.
3. The dataset has more number of closed, medium maintenance priority and industrial buildings.
4. The houses that are green certified are almost 3 times the houses that are not.
5. Majority of the buildings have very low maintenance resolution time.
6. Comercial buildings tend to have more variation in electricity consumption while for others it is somewhat concentrated.
7. Overall the data doesn't show any significant pattern which changing any of the parameters hence the data is very well balanced and it is hard to predict the electricity consumption.

(b).

Used Umap reducer from sk learn to get -



After dimensionality reduction the data set is separable into 2 clusters majorly.

(c).

Preprocessing:

1. Used label encoder from sk learn to encode the values for categorical features.
2. Used standard scaler from sk learn to normalize the data for numerical features.

Applied Linear Regression using SK Learn library. Trained the model on train set and prediction on both train set and test set. Reported the results (MSE, RMSE, R2 score, Adjusted R2 score, MAE) on the train and test dataset using sklearn.

```
Train Data Metrics:
MSE: 24475013.1685
RMSE: 4947.2228
R2: 0.0139
Adjusted R2: -0.0011
MAE: 4006.3285

Test Data Metrics:
MSE: 24278016.1557
RMSE: 4927.2727
R2: 0.0000
Adjusted R2: -0.0641
MAE: 3842.4093
```

(d).

Performed RFE using SK learn (putting linear regression as the model), and the most important features came out to be building type, green certified, and number of residents. Trained model using only 3 features and reported the results using sk learn library.

```
Selected Features using RFE:
Index(['Building_Type', 'Green_Certified', 'Number_of_Residents'], dtype='object')
Train Data Metrics:
MSE: 24569032.9069
RMSE: 4956.7159
R2: 0.0101
Adjusted R2: 0.0072
MAE: 4006.4734

Test Data Metrics:
MSE: 23941409.0630
RMSE: 4892.9959
R2: 0.0139
Adjusted R2: 0.0019
MAE: 3813.9481
```

Results Comparision with part (c):

1. Train Set:
   a. MSE: It increased slightly.
   b. RMSE: It also increased slightly.
   c. R2: It decreased.
   d. Adjusted R2: It become positive(increased)
   e. MAE: It nearly remained unchanged

2. Test Set:
   a. MSE: It decreased slightly.
   b. RMSE: It also decreased slightly.
   c. R2: It Increased very significantly.
   d. Adjusted R2: It become positive(increased)
   e. MAE: It decreased.

(e).

Used the OneHotEncoder library from sklearn on the original dataset to encode the values for categorical features. Scaled the data using a standard scaler. Imported Ridge model from sklearn and trained on the train data using alpha = 0.1. Made predictions for both the train set and the test set. Computed all the result values using the sklearn library as done above.

```
Train Data Metrics:
MSE: 24188934.3377
RMSE: 4918.2247
R2: 0.0254
Adjusted R2: 0.0066
MAE: 3976.7359

Test Data Metrics:
MSE: 24128285.0391
RMSE: 4912.0551
R2: 0.0062
Adjusted R2: -0.0759
MAE: 3797.5125
```

Results Comparision with part (c):

1. Train Set:
   a. MSE: It decreased slightly.
   b. RMSE: It also decreased slightly.
   c. R2: It Increased significantly.
   d. Adjusted R2: It become positive(increased)
   e. MAE: It decreased.

2. Test Set:
   a. MSE: It decreased.
   b. RMSE: It also decreased slightly.
   c. R2: It Increased very significantly.
   d. Adjusted R2: It remains approximately unchanged.
   e. MAE: It decreased.

(f).

Used the one hot encoded dataset from part (e) and performed ICA using SKlearn and tried all the component values given in assignment - 4,5,6,8.

ICA Components: 4
Train Data Metrics:
MSE: 24794328.8383
RMSE: 4979.3904
R2: 0.0011
Adjusted R2: -0.0030
MAE: 4009.1470

Test Data Metrics:

MSE: 24354011.5513
RMSE: 4934.9784
R2: -0.0031
Adjusted R2: -0.0195
MAE: 3841.5961

ICA Components: 5
Train Data Metrics:
MSE: 24642116.2028
RMSE: 4964.0826
R2: 0.0072
Adjusted R2: 0.0022
MAE: 4016.2416

Test Data Metrics:
MSE: 24521135.1562
RMSE: 4951.8820
R2: -0.0100
Adjusted R2: -0.0307
MAE: 3855.2177

ICA Components: 6
Train Data Metrics:
MSE: 24639512.0472
RMSE: 4963.8203
R2: 0.0073
Adjusted R2: 0.0013
MAE: 4016.5231

Test Data Metrics:
MSE: 24477194.7327
RMSE: 4947.4433
R2: -0.0082
Adjusted R2: -0.0331
MAE: 3849.9652

ICA Components: 8
Train Data Metrics:
MSE: 24626103.5104
RMSE: 4962.4695
R2: 0.0078
Adjusted R2: -0.0002
MAE: 4021.3877

Test Data Metrics:
MSE: 24544800.0329
RMSE: 4954.2709
R2: -0.0110
Adjusted R2: -0.0445
MAE: 3861.5383

Results Comparision while increasing the number of components:

1. Train Set:
   a. MSE: It decreased.
   b. RMSE: It also decreased.
   c. R2: It Increased.
   d. Adjusted R2: It increased then decreased
   e. MAE: It nearly remained unchanged

2. Test Set:
   a. MSE: It nearly remained unchanged
   b. RMSE: It Increased.
   c. R2: It Increased very significantly.
   d. Adjusted R2: It decreased.
   e. MAE: It remains approximately unchanged.

(g).

Used the preprocessed data from part (c) and applied ElasticNet regularisation by importing ElasticNet from SKlearn.

Parameters: l1 ratio = 0.5

Alpha: 0.01
Train Data Metrics:
MSE: 24475013.1685
RMSE: 4947.2228
R2: 0.0139
Adjusted R2: -0.0011
MAE: 4006.3285

Test Data Metrics:
MSE: 24278016.1557
RMSE: 4927.2727
R2: 0.0000
Adjusted R2: -0.0641

MAE: 3842.4093

Alpha: 0.05
Train Data Metrics:
MSE: 24512724.4045
RMSE: 4951.0327
R2: 0.0124
Adjusted R2: -0.0027
MAE: 4001.8522

Test Data Metrics:
MSE: 24230582.3136
RMSE: 4922.4569
R2: 0.0020
Adjusted R2: -0.0620
MAE: 3834.9349

Alpha: 0.1
Train Data Metrics:
MSE: 24560029.5779
RMSE: 4955.8077
R2: 0.0105
Adjusted R2: -0.0046
MAE: 4001.8452

Test Data Metrics:
MSE: 24235440.5364
RMSE: 4922.9504
R2: 0.0018
Adjusted R2: -0.0622
MAE: 3832.6544

Alpha: 0.5
Train Data Metrics:
MSE: 24597783.9741
RMSE: 4959.6153
R2: 0.0090
Adjusted R2: -0.0061
MAE: 4002.1717

Test Data Metrics:
MSE: 24247450.1195
RMSE: 4924.1700
R2: 0.0013

Adjusted R2: -0.0627
MAE: 3832.6183

Alpha: 1
Train Data Metrics:
MSE: 24626971.8232
RMSE: 4962.5570
R2: 0.0078
Adjusted R2: -0.0073
MAE: 4002.6146

Test Data Metrics:
MSE: 24259288.5170
RMSE: 4925.3719
R2: 0.0008
Adjusted R2: -0.0632
MAE: 3833.5506

Alpha: 5
Train Data Metrics:
MSE: 24649833.7857
RMSE: 4964.8599
R2: 0.0069
Adjusted R2: -0.0083
MAE: 4002.9935

Test Data Metrics:
MSE: 24269644.4204
RMSE: 4926.4231
R2: 0.0004
Adjusted R2: -0.0637
MAE: 3834.3655

Alpha: 10
Train Data Metrics:
MSE: 24668101.3070
RMSE: 4966.6992
R2: 0.0061
Adjusted R2: -0.0090
MAE: 4003.4046

Test Data Metrics:
MSE: 24278461.6108
RMSE: 4927.3179

R2: 0.0000
Adjusted R2: -0.0641
MAE: 3835.2504

Alpha: 50
Train Data Metrics:
MSE: 24682983.2777
RMSE: 4968.1972
R2: 0.0055
Adjusted R2: -0.0096
MAE: 4003.7801

Test Data Metrics:
MSE: 24285947.5264
RMSE: 4928.0775
R2: -0.0003
Adjusted R2: -0.0644
MAE: 3835.9442

Alpha: 100
Train Data Metrics:
MSE: 24695318.3763
RMSE: 4969.4384
R2: 0.0050
Adjusted R2: -0.0101
MAE: 4004.1500

Test Data Metrics:
MSE: 24292335.0363
RMSE: 4928.7255
R2: -0.0006
Adjusted R2: -0.0647
MAE: 3836.5267

Results Comparision while increasing the value of alpha:

1. Train Set:
   a. MSE: It increases slightly.
   b. RMSE: It also increases.
   c. R2: It decreases.
   d. Adjusted R2: It decreases.
   e. MAE: It increases slightly.

2. Test Set:
    a. MSE: It nearly remained unchanged
    b. RMSE: It nearly remained unchanged
    c. R2: It nearly remained unchanged
    d. Adjusted R2: It nearly remained unchanged
    e. MAE: It nearly remained unchanged

(h).

Used preprocessed data from part (c) and applied `GradientBoostingRegressor` Directly from sklearn-ensemble and calculated results using the common function.

```
Train Data Metrics:
MSE: 14926446.2573
RMSE: 3863.4759
R2: 0.3986
Adjusted R2: 0.3895
MAE: 3092.7482

Test Data Metrics:
MSE: 24405496.6167
RMSE: 4940.1920
R2: -0.0052
Adjusted R2: -0.0697
MAE: 3813.6305
```

Results Comparision with part (c):

1. Train Set:
    a. MSE: It decreased very significantly.
    b. RMSE: It decreased very significantly.
    c. R2: It Increased significantly.
    d. Adjusted R2: It become positive(increased significantly)
    e. MAE: It decreased slightly.

1. Test Set:
    a. MSE: It increased slightly.
    b. RMSE: It increased slightly.
    c. R2: It became negative.
    d. Adjusted R2: It remains approximately unchanged.
    e. MAE: It decreased.

Results Comparision with part (g):

2. Train Set:

a. MSE: It decreased very significantly.
b. RMSE: It decreased very significantly.
c. R2: It Increased significantly.
d. Adjusted R2: It become positive(increased significantly)
e. MAE: It decreased slightly.

3. Test Set:
   a. MSE: It decreased slightly.
   b. RMSE: It also decreased.
   c. R2: It became negative.
   d. Adjusted R2: It decreased..
   e. MAE: It decreased.