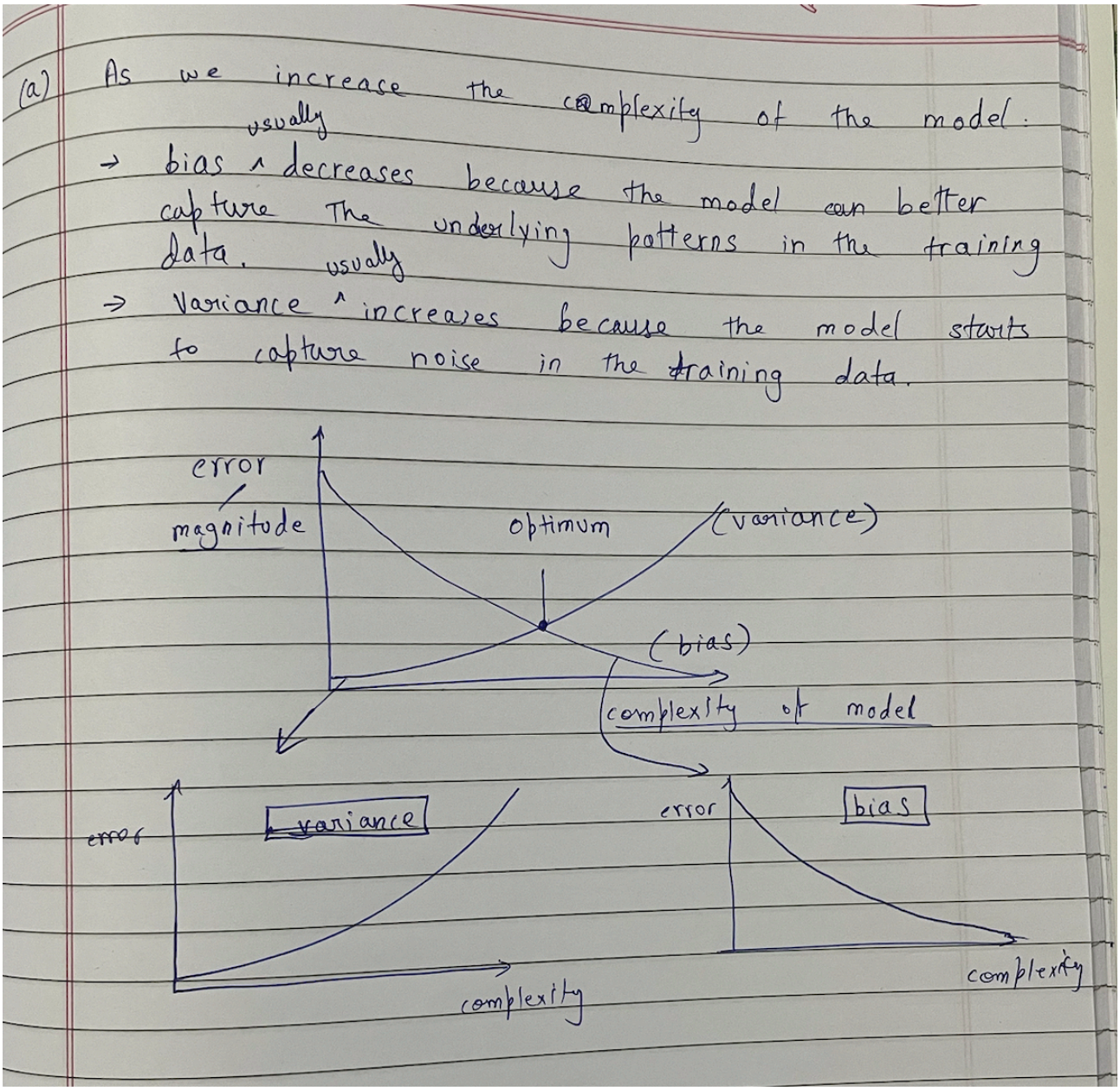
Report: ML Assignment – 1

Nimit Panwar, 2022324

Section A

Ans 1 :



As we increase the complexity of the model, bias usually decreases as the model can better capture the underlying patterns in the training data. Variance usually increases because the model starts to capture noise in the training data
(graph is similar to the one done in Lecture 7)

Ans 2:

(b) True positive = 200
False Positive = 20
True negative = 730
False negative = 50

$$\text{Precision} = \frac{\text{True positive}}{\text{true positive + false positive}} = \frac{200}{200+20}$$

$$= \frac{20}{22} = \frac{10}{11} = \underline{0.909}$$

$$\text{Sensitivity} = \frac{\text{true positive}}{\text{true positive + false negative}} = \frac{200}{200+50}$$

$$= \frac{200}{250} = \underline{0.8}$$

$$\text{Specificity} = \frac{\text{True negative}}{\text{true negative + false positive}} = \frac{730}{730+20} = \underline{0.973}$$

$$\text{Negative Predictive Value} = \frac{\text{true negative}}{\text{true negative + false negatives}}$$

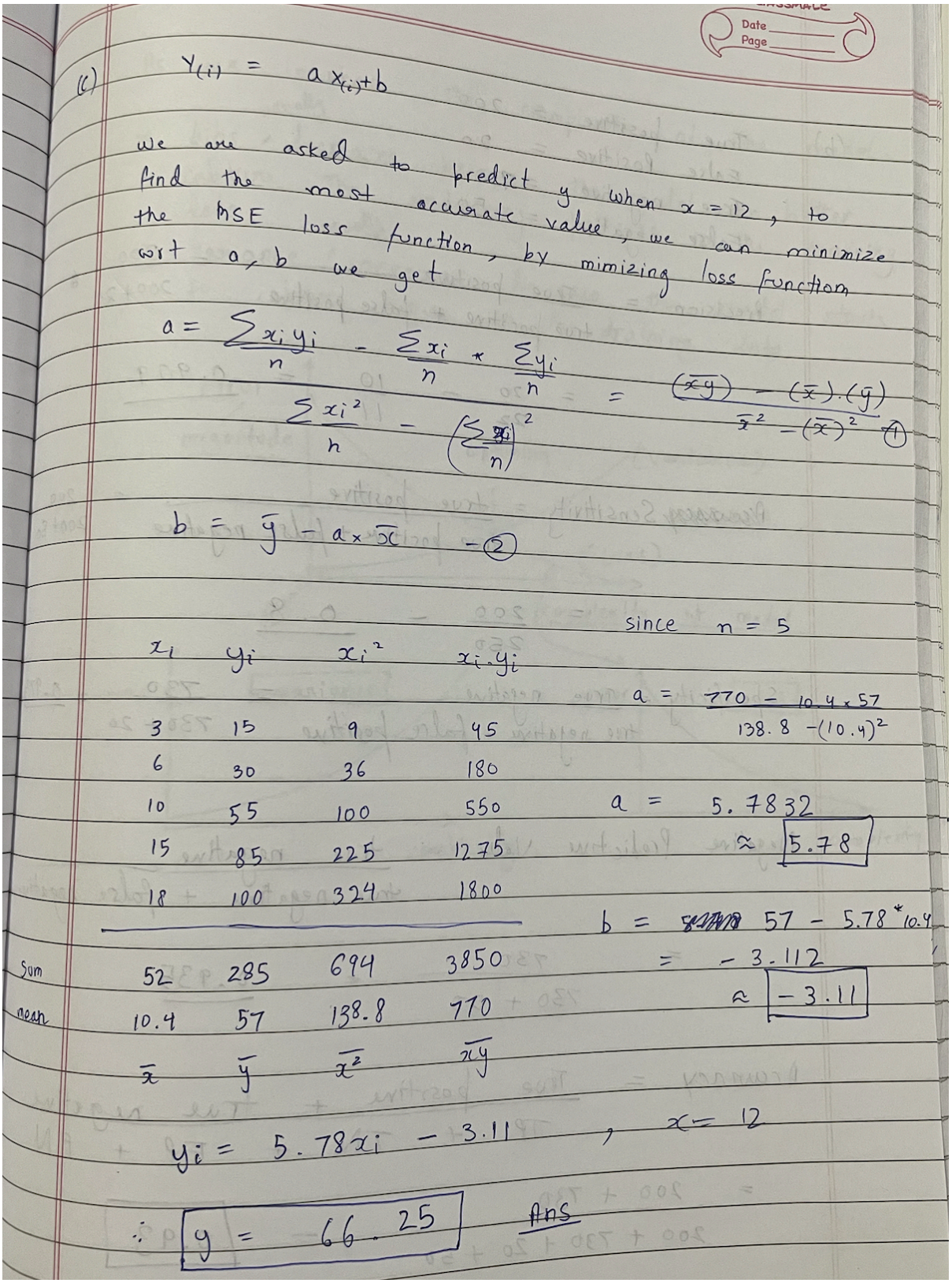
$$= \frac{730}{730+50} = \underline{0.935}$$

$$\text{Accuracy} = \frac{\text{True positive + true negative}}{TP + TN + FP + FN}$$

$$= \frac{200+730}{200+730+20+50} = \boxed{0.93}$$

Precision = TP/(TP+FP) = 200/220 = 0.909
Sensitivity = TP/(TP+FN) = 200/250 = 0.8
Specificity = TN/(TN+FP) = 730/750 = 0.973
Negative Predictive Value = TN/(TN+FN) = 730/780 = 0.935
Accuracy = TN+TP/(TN+FN+TP+FP) = 930/1000 = 0.93

Ans 3 :



(c) $Y_{(i)} = a x_{(i)} + b$

we are asked to predict y when $x = 12$, to find the most accurate value, we can minimize the MSE loss function, by mimizing loss function wrt a, b we get

$$a = \frac{\sum x_i y_i}{n} - \frac{\sum x_i}{n} * \frac{\sum y_i}{n} = \frac{(\overline{xy}) - (\bar{x}).(\bar{y})}{\bar{x}^2 - (\bar{x})^2} \quad -(1)$$

$$\frac{\sum x_i^2}{n} - \left(\frac{\sum x_i}{n}\right)^2$$

$$b = \bar{y} - a \times \bar{x} \quad -(2)$$

since $n = 5$

	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
mean	10.4	57	138.8	770
	$\bar{x}$	$\bar{y}$	$\bar{x^2}$	$\bar{xy}$

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

$$a = 5.7832 \approx \boxed{5.78}$$

$$b = 57 - 5.78 * 10.4$$

$$= -3.112 \approx \boxed{-3.11}$$

$$y_i = 5.78 x_i - 3.11 \quad , \quad x = 12$$

$$\therefore \boxed{y = 66.25} \quad \text{Ans}$$

We will have to calculate a and b.
First we will calculate mean of x,y,x^2 and xy as shown in the table.

we will then calculate a and b by substituting it into the formula

a = (summation (xi x yi)/n − summation(xi)/n x summation(yi)/n)/
summation(xi^2)/n − (summation(xi)/n)^2

And b = mean(y) − a x mean(x)

We get a = 5.78 and b = −3.11

We now have our equation as

y = 5.78x −3.11

we put x=12, and get y = 66.25

Ans 4 :

Height (cm)	Weight (kg)	Overweight (Y)
160	55	0
170	60	0
180	80	1
190	95	1

We want to classify whether someone is overweight based on height and weight.

Lets compare two models using Linear Regression (f1) and Logistic Regression (f2)

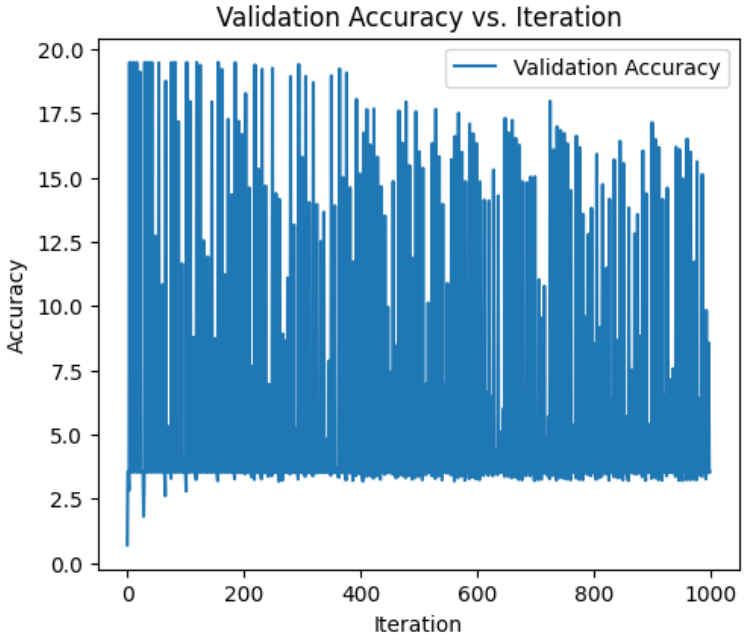
Linear Regression (f1) might have a lower empirical risk on the training set because it fits a line, but it's not suitable for binary classification and may overfit, leading to poor generalization.

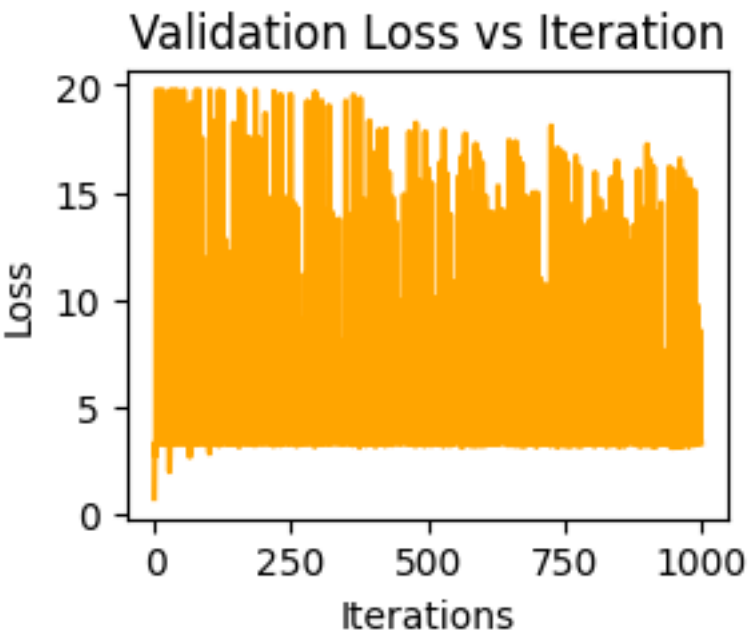
Logistic Regression (f2), designed for classification, could have a higher empirical risk on the training data but generalizes better to unseen data due to its appropriate handling of binary outcomes.

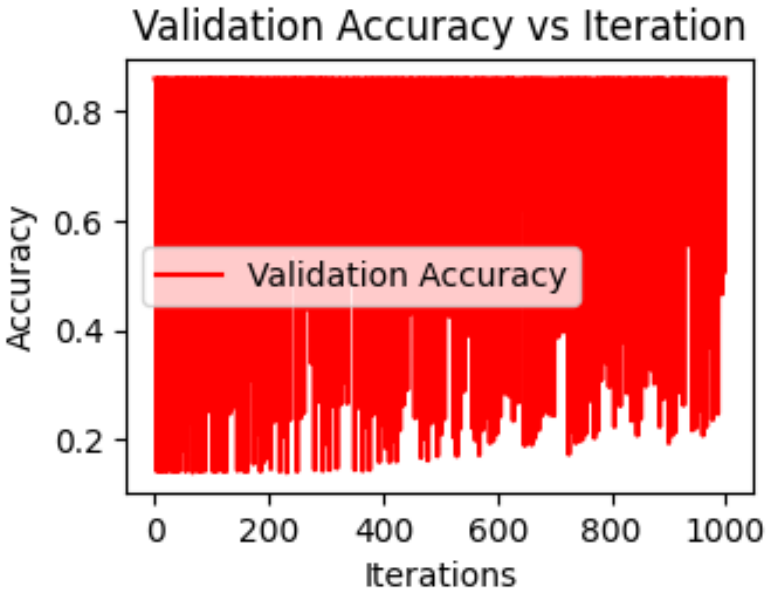
So, f1 may perform better on training data but generalize worse than f2 .

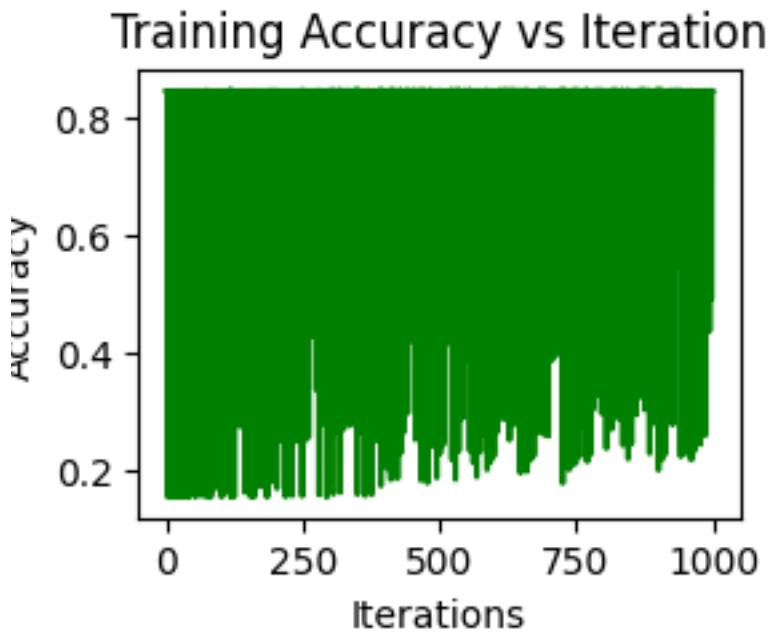
Section B

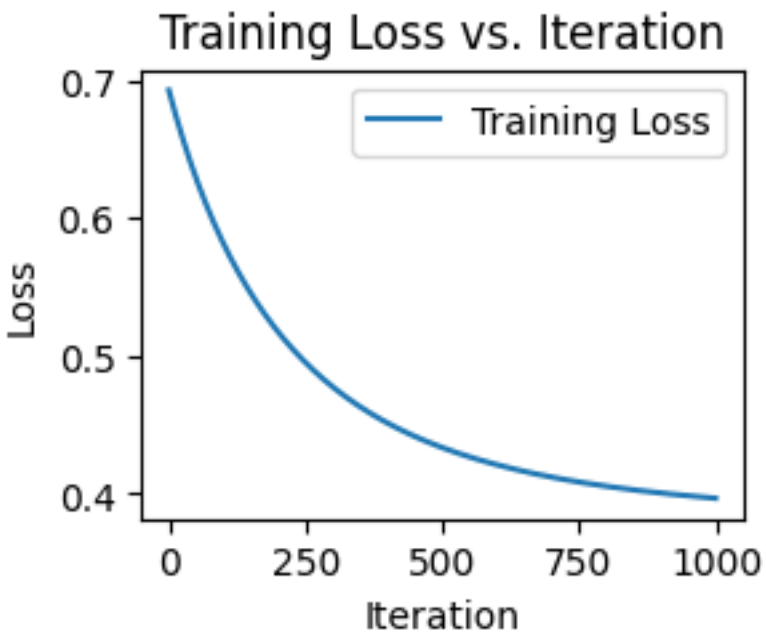
A) In the data preprocessing step, any missing values in the dataset were filled with the average value of their respective columns to ensure the data was complete

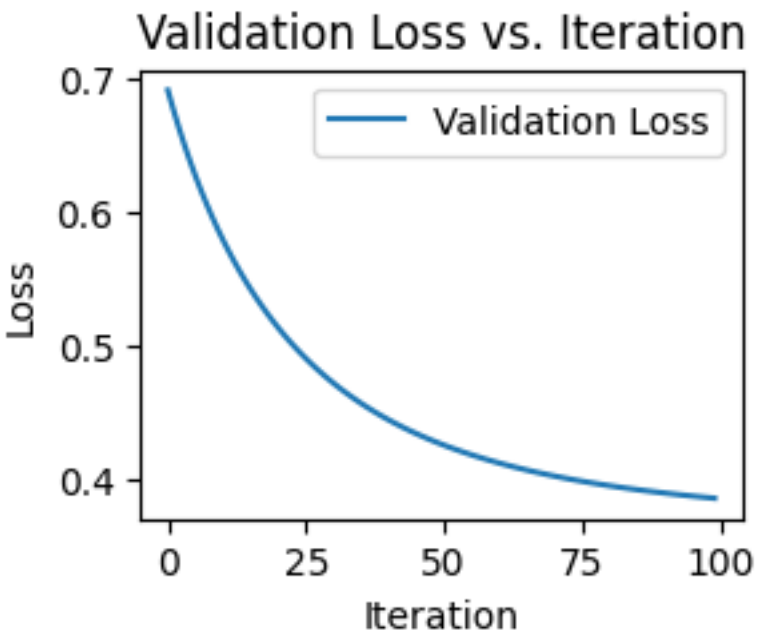


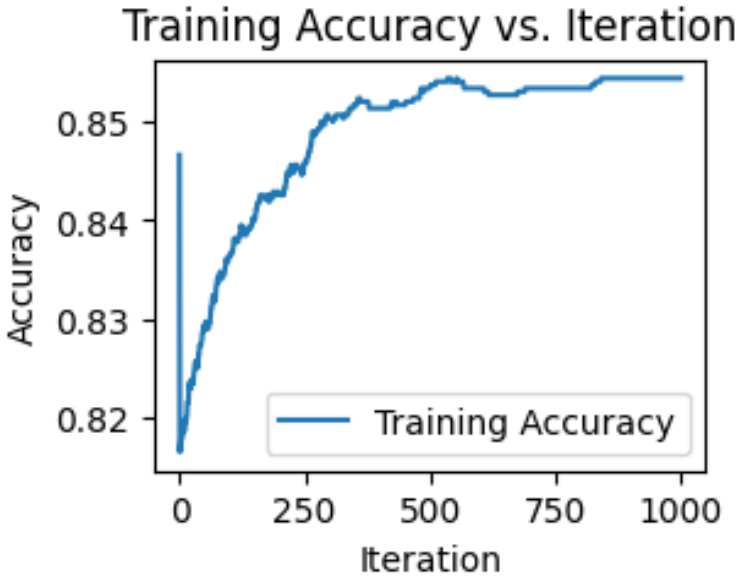


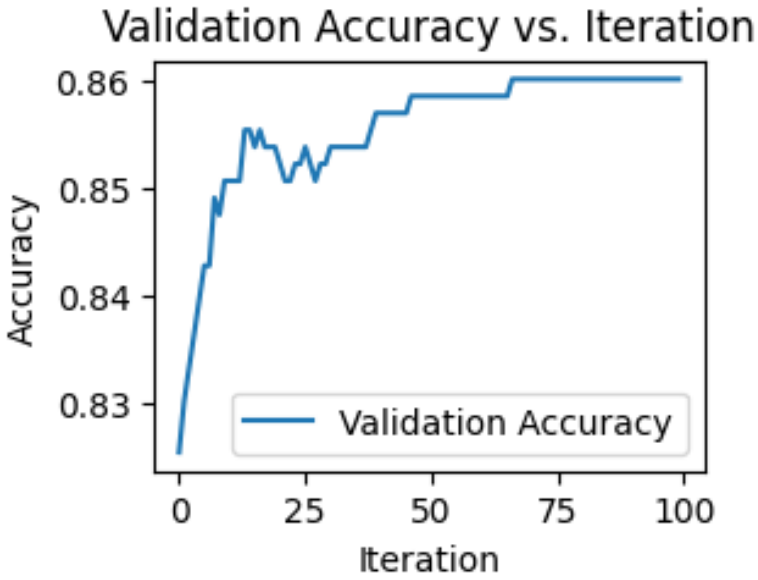













```
Final Metrics:
Final Training Loss: 3.5400
Final Validation Loss: 3.2584
Final Training Accuracy: 0.8463
Final Validation Accuracy: 0.8585
```

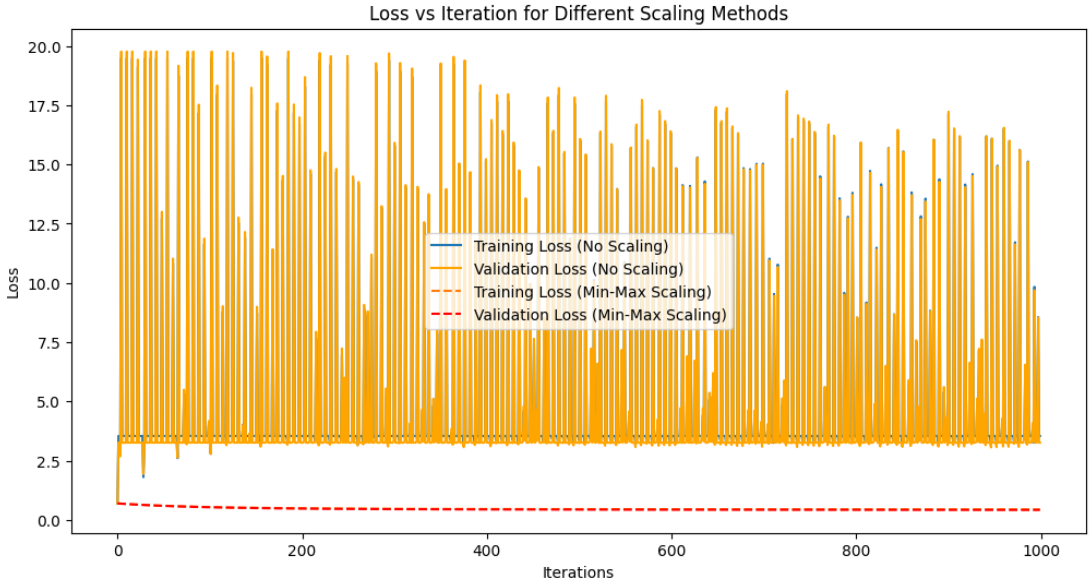
At the start of training, the training loss is relatively high, indicating that the model is initially making large errors in its predictions. By the end of the training process, the training loss is still high and fluctuating, the model has not effectively minimized the training loss.
The validation loss at the beginning is also high, similar to the training loss, which is expected as the model has not yet learned

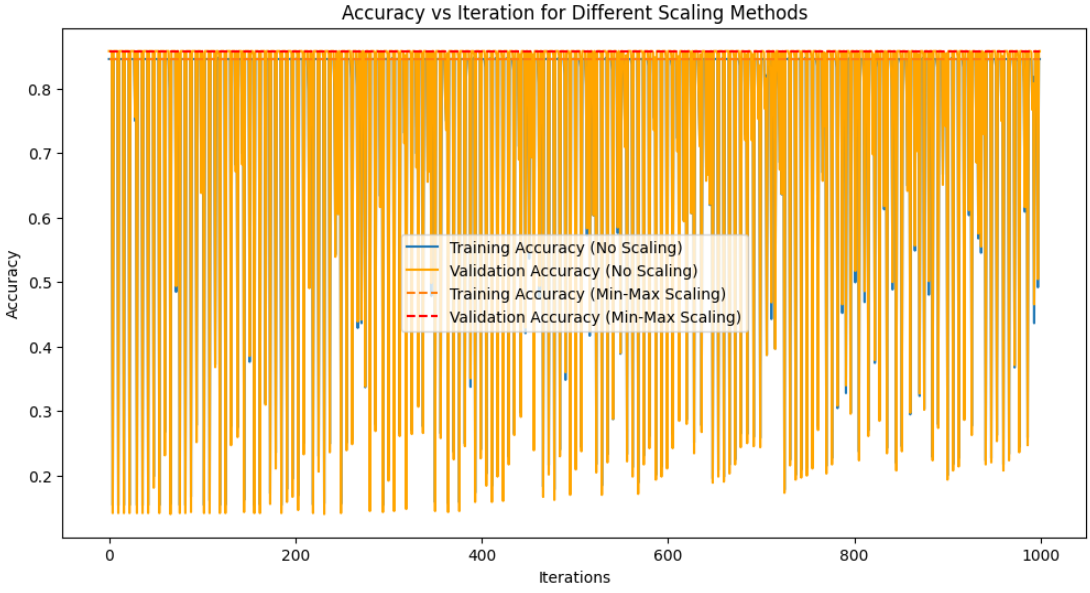
from the data. The validation loss at the end is similar to training curve.
Overall both the training and validation loss plots show a clear downward trend from the beginning to the end.
The stability of the model is not good in the current model.

B)

Final Metrics (No Scaling):
Final Training Loss: 3.5400
Final Validation Loss: 3.2584
Final Training Accuracy: 0.8463
Final Validation Accuracy: 0.8585

Final Metrics (Min-Max Scaling):
Final Training Loss: 0.4413
Final Validation Loss: 0.4190
Final Training Accuracy: 0.8463
Final Validation Accuracy: 0.8585

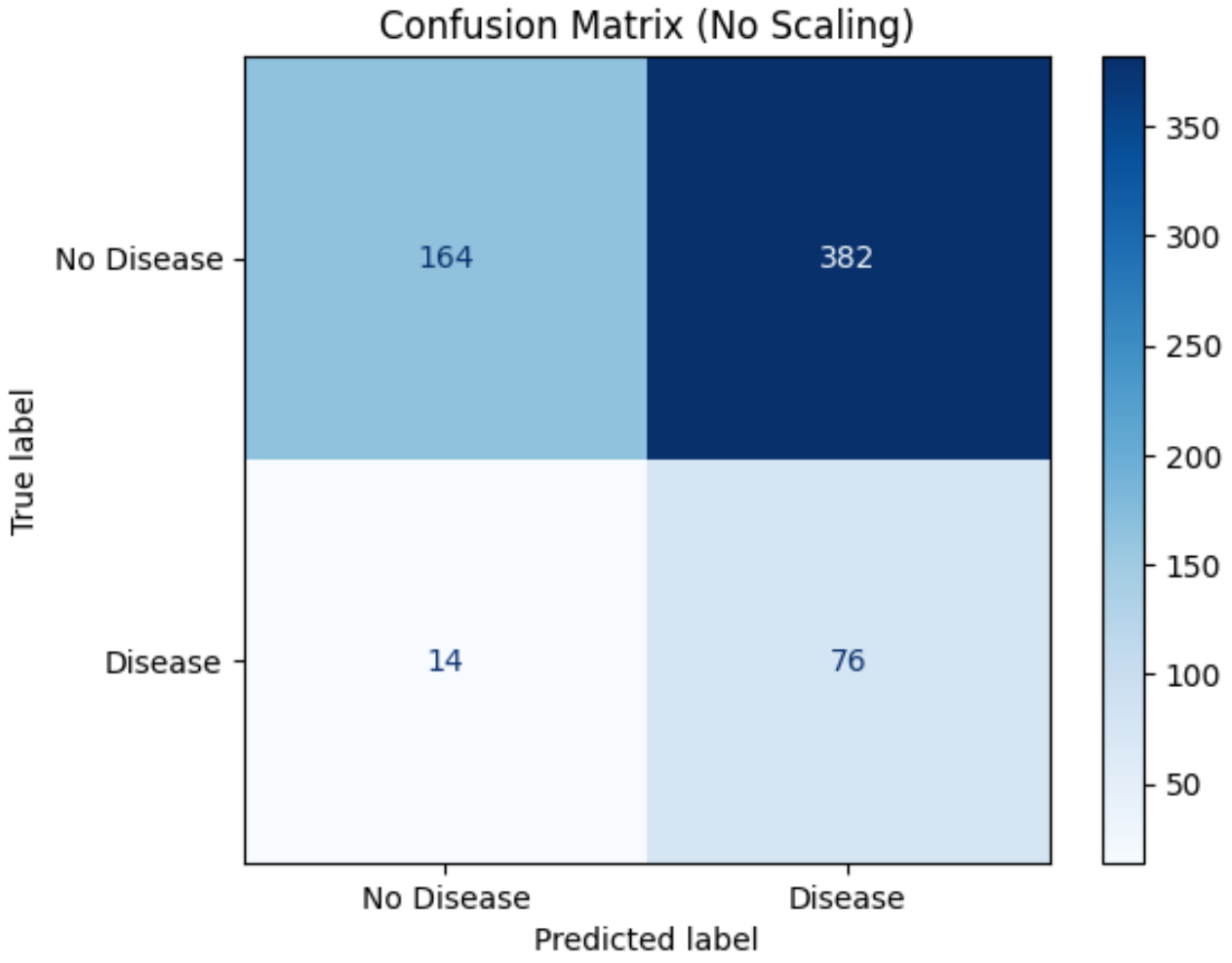




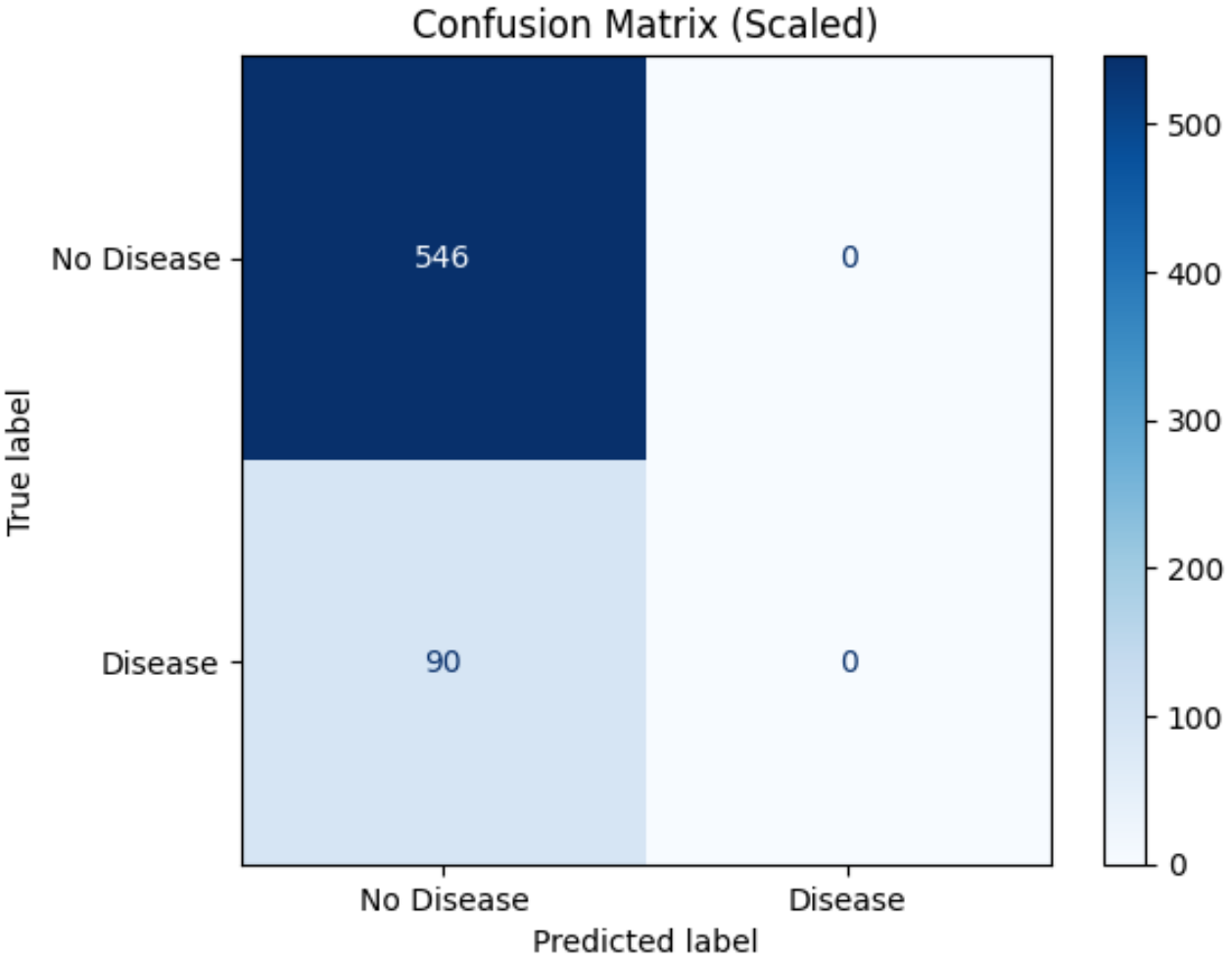
Min-Max Scaling has improved the learning by reducing both training and validation losses.
Without scaling, the training loss is 3.5400, but with scaling, it drops to 0.4413.
Similarly, the validation loss decreases from 3.2584 to 0.4190 with scaling.
This shows that scaling helps the model generalize better to new data.
However, the final training and validation accuracies remain the same (0.8463 and 0.8585, respectively) whether scaling is applied or not. This means that while scaling improves the learning process, it doesn't change the model's overall accuracy.

C)

```
Confusion Matrix (No Scaling):
[[164 382]
 [ 14  76]]

Precision (No Scaling): 0.1659
Recall (No Scaling): 0.8444
F1 Score (No Scaling): 0.2774
ROC-AUC Score (No Scaling): 0.5775
```



```
Confusion Matrix (Scaled):
[[546   0]
 [ 90   0]]

Precision (Scaled): 1.0000
Recall (Scaled): 0.0000
F1 Score (Scaled): 0.0000
ROC-AUC Score (Scaled): 0.7210
```



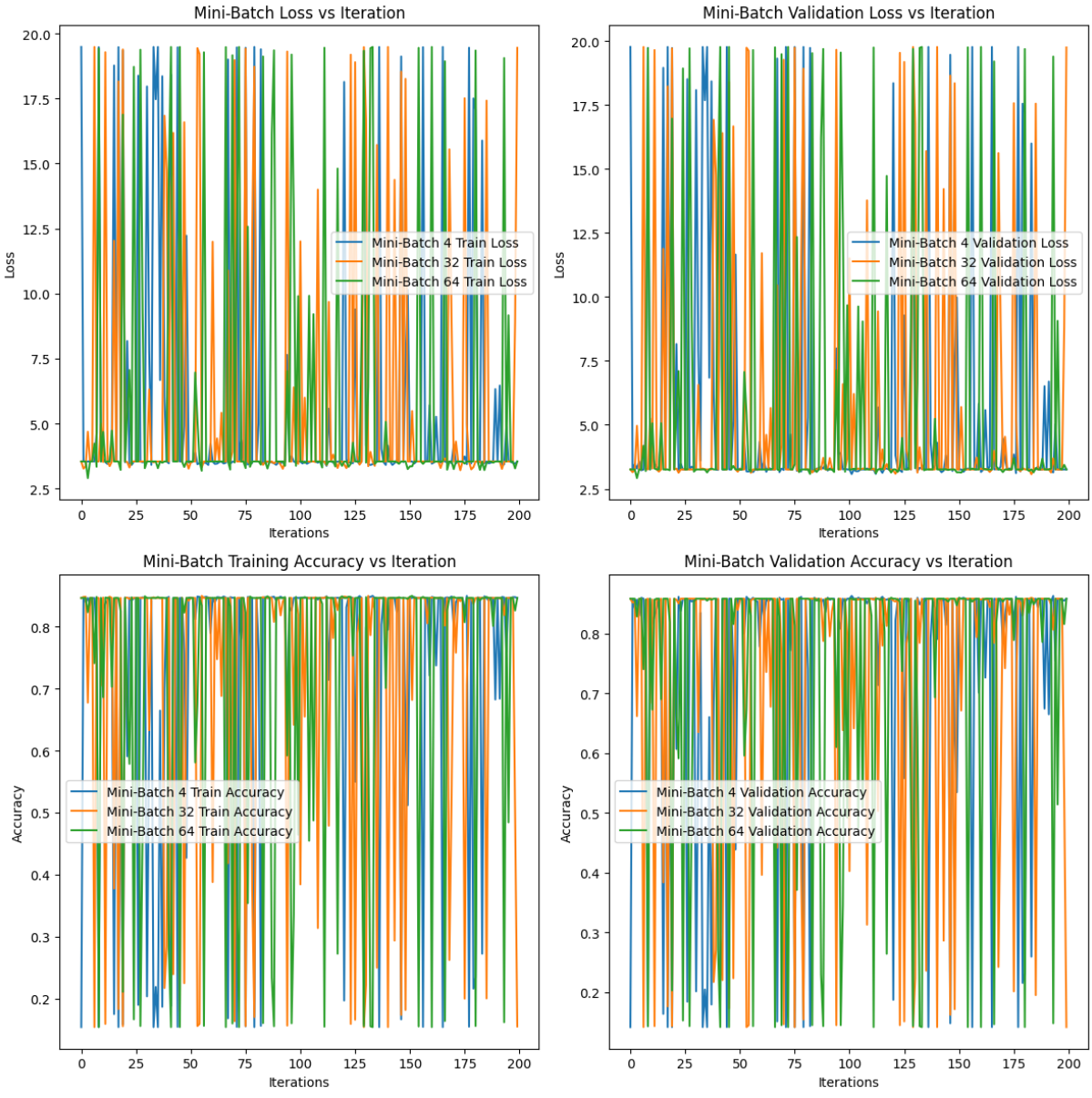
Without scaling, the model identifies many positives but with low accuracy, resulting in a low precision (0.1659) and a high recall (0.8444). The F1 score (0.2774) and ROC-AUC score (0.5775) indicate poor balance and limited class distinction. With scaling, the model correctly identifies all true negatives but fails to predict any positives, resulting in perfect but misleading precision (1.0000), zero recall (0.0000), and an F1 score of 0.0000. The ROC-AUC score improves to 0.7210, suggesting better class distinction but still failing to predict positives. Both scenarios indicate the need for further tuning for improvement.

D)
Mini-Batch Gradient Descent Results:

Batch Size: 4
Train Loss: 3.5400499070478255
Validation Loss: 3.258375131681197
Train Accuracy: 0.8462575859743763
Validation Accuracy: 0.8584905660377359

Batch Size: 32
Train Loss: 4.179755050089128
Validation Loss: 4.376032789875426
Train Accuracy: 0.7623061362103843
Validation Accuracy: 0.75

Batch Size: 64
Train Loss: 13.398016818421464
Validation Loss: 13.150622264614547
Train Accuracy: 0.33007417397167904
Validation Accuracy: 0.33962264150943394

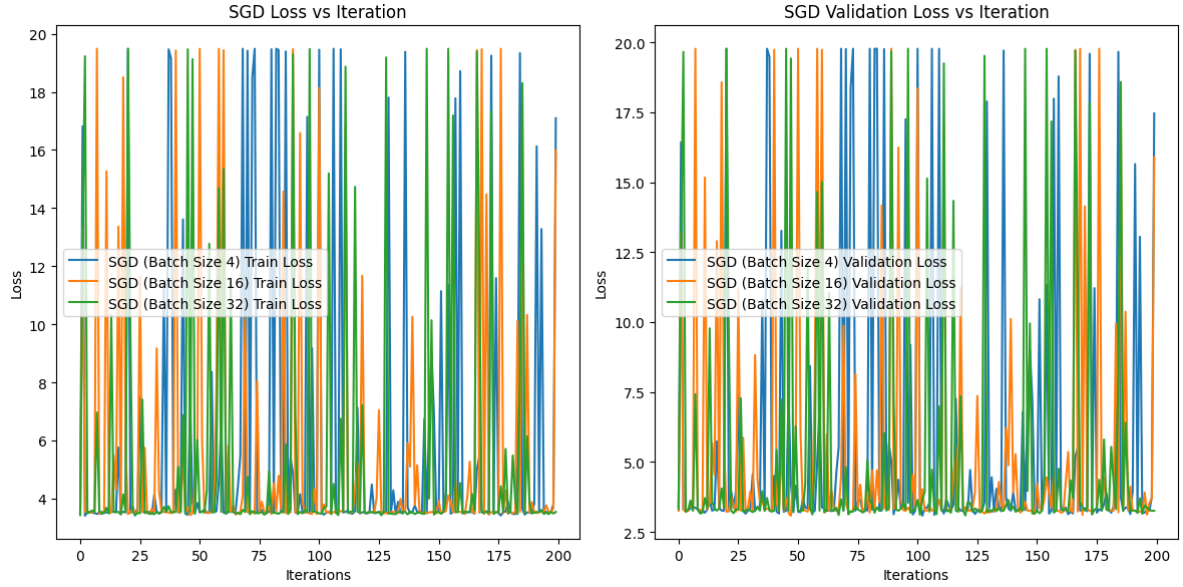


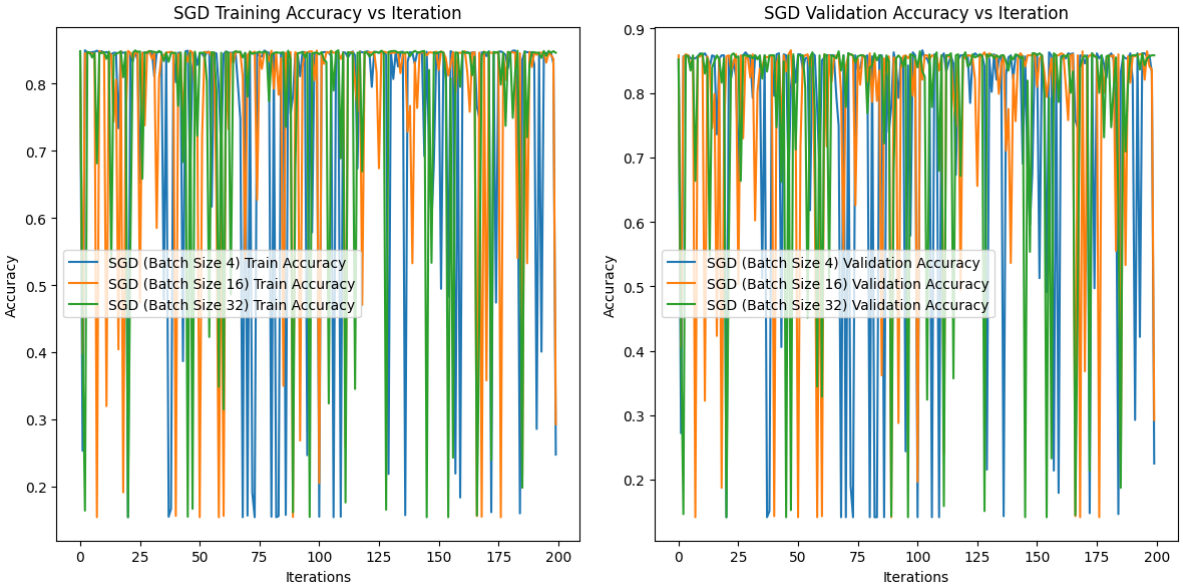
Stochastic Gradient Descent Results:

Batch Size: 4
Train Loss: 18.309970563949488
Validation Loss: 18.274166862248503
Train Accuracy: 0.1952124072825354
Validation Accuracy: 0.19811320754716982

Batch Size: 16
Train Loss: 3.5384294051352296
Validation Loss: 3.1533333162509907
Train Accuracy: 0.8462575859743763
Validation Accuracy: 0.860062893081761

Batch Size: 32
Train Loss: 14.98196315305271
Validation Loss: 14.59824067662144
Train Accuracy: 0.33007417397167904
Validation Accuracy: 0.33962264150943394





Mini-Batch Gradient Descent:

Batch Size: 4
Fast convergence due to frequent updates and more noise in the updates, which can lead to fluctuations in the loss and accuracy curves. It has achieved high accuracy and relatively low loss.

Batch Size: 32
Slower convergence compared to smaller batch sizes And more stable updates, but the larger batch size may lead to slower learning. It has Lower accuracy and higher loss, indicating potential underfitting.

Batch Size: 64
Similar to batch size 32, with slower convergence. Stable updates, but may not capture the noise of the data as effectively. It has high accuracy and low loss, indicating good generalization.

Stochastic Gradient Descent:

Batch Size: 4
Very fast convergence due to updates after each sample.
High noise in the updates, leading to significant fluctuations.
High accuracy and low loss, but the noise can make the training process less predictable.

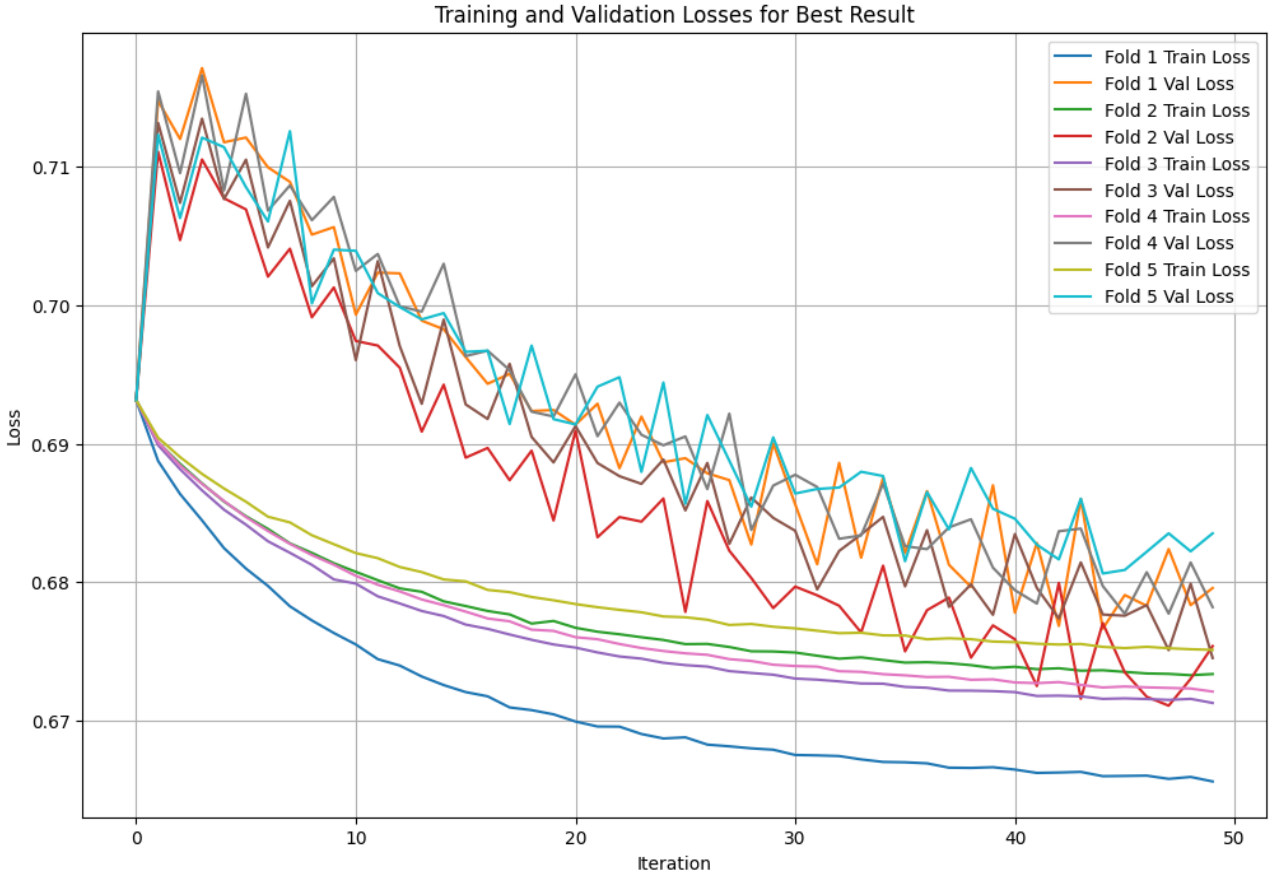
Batch Size: 16
Balanced convergence speed, faster than larger batch sizes but slower than batch size 4.
Moderate noise, providing a balance between stability and convergence speed.
High accuracy and low loss, similar to batch size 4.

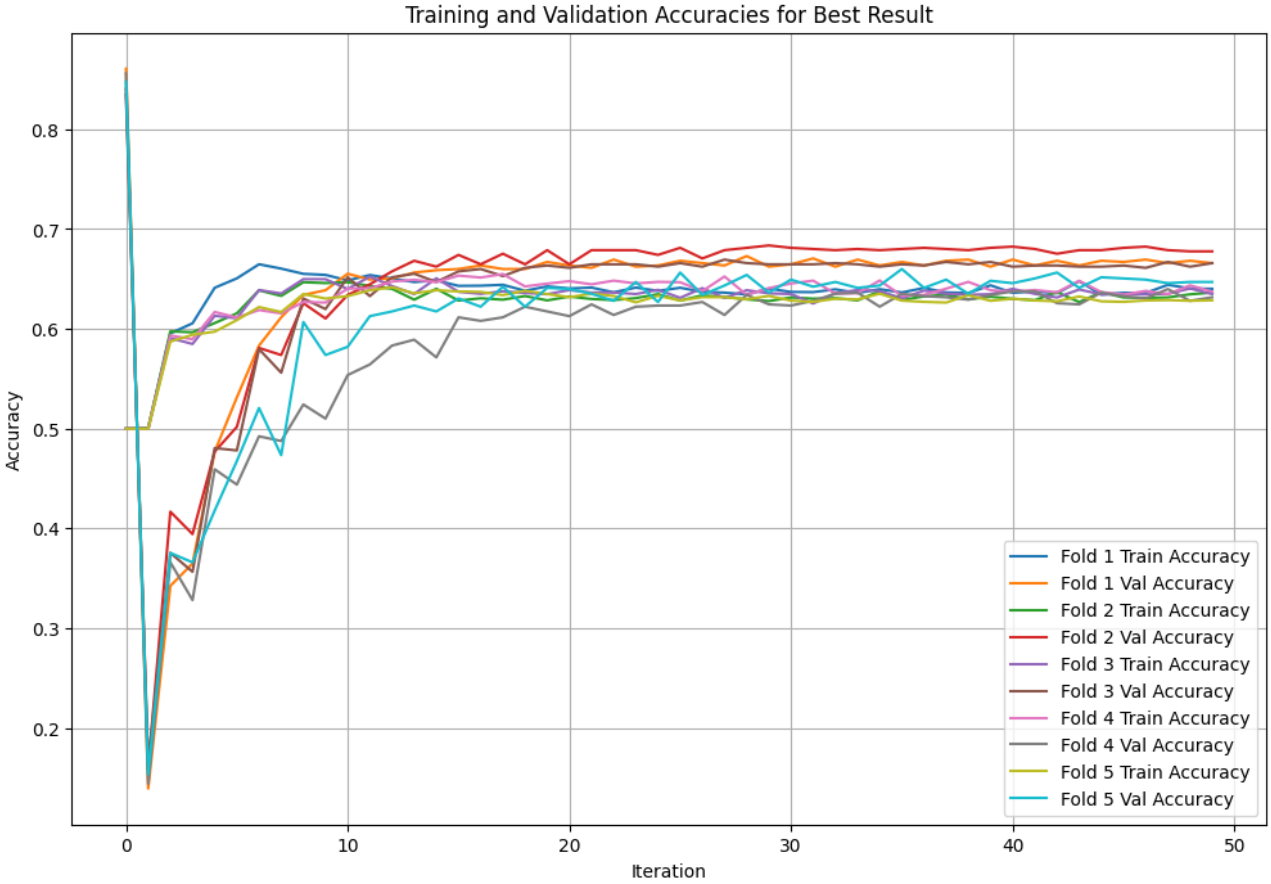
Batch Size: 32
Slower convergence compared to smaller batch sizes.
More stable updates, but slower learning.
Lower accuracy and higher loss, indicating potential underfitting.

In summary, Smaller batch sizes offer faster convergence but with higher noise, while larger batch sizes provide more stable updates but may converge slower.

E)

```
Running k-fold cross-validation with learning rate: 0.001
Running k-fold cross-validation with learning rate: 0.05
Running k-fold cross-validation with learning rate: 0.01
Running k-fold cross-validation with learning rate: 0.5
Running k-fold cross-validation with learning rate: 0.1
Running k-fold cross-validation with learning rate: 1

We get the Best Result for : Learning Rate: 0.5

Validation Accuracy: 0.6621 ± 0.0125
Validation Precision: 0.2453 ± 0.0278
Validation Recall: 0.5917 ± 0.0553
Validation F1 Score: 0.3464 ± 0.0352

Results for each fold of the best learning rate:

Fold 1:
Accuracy: 0.6682
Precision: 0.2844
Recall: 0.6643
F1 Score: 0.3983

Fold 2:
Accuracy: 0.6423
Precision: 0.2337
Recall: 0.6423
F1 Score: 0.3427

Fold 3:
Accuracy: 0.6635
Precision: 0.2391
Recall: 0.5462
F1 Score: 0.3326

Fold 4:
Accuracy: 0.6800
Precision: 0.2036
Recall: 0.5185
F1 Score: 0.2924

Fold 5:
Accuracy: 0.6564
Precision: 0.2658
Recall: 0.5874
F1 Score: 0.3660
```


Training and Validation Losses for Best Result
Fold 1 Train Loss
Fold 1 Val Loss
Fold 2 Train Loss
Fold 2 Val Loss
Fold 3 Train Loss
Fold 3 Val Loss
Fold 4 Train Loss
Fold 4 Val Loss
Fold 5 Train Loss
Fold 5 Val Loss
Loss
0.71
0.70
0.69
0.68
0.67
0
10
20
30
40
50
Iteration


Training and Validation Accuracies for Best Result
Fold 1 Train Accuracy
Fold 1 Val Accuracy
Fold 2 Train Accuracy
Fold 2 Val Accuracy
Fold 3 Train Accuracy
Fold 3 Val Accuracy
Fold 4 Train Accuracy
Fold 4 Val Accuracy
Fold 5 Train Accuracy
Fold 5 Val Accuracy
Accuracy
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
10
20
30
40
50
Iteration

Stability and Variance of the Model’s Performance Across Different Folds

Validation Accuracy –
The low standard deviation indicates that the model's validation accuracy is relatively stable across different folds, with minimal variance.

Validation Precision Mean –
The precision has a higher standard deviation compared to accuracy, suggesting more variability in the model's ability to correctly identify positive instances across different folds.

Validation Recall –
The recall shows the highest standard deviation among the metrics, indicating significant variability in the model's ability to capture all relevant positive instances across different folds.
Validation F1 Score Mean –

The F1 score, which balances precision and recall, has a moderate standard deviation, suggesting some variability but not as pronounced as recall.

The model's performance, particularly in terms of validation accuracy, is relatively stable across different folds. However, there is a variation in precision, recall, and F1 score, with recall showing the highest variance.

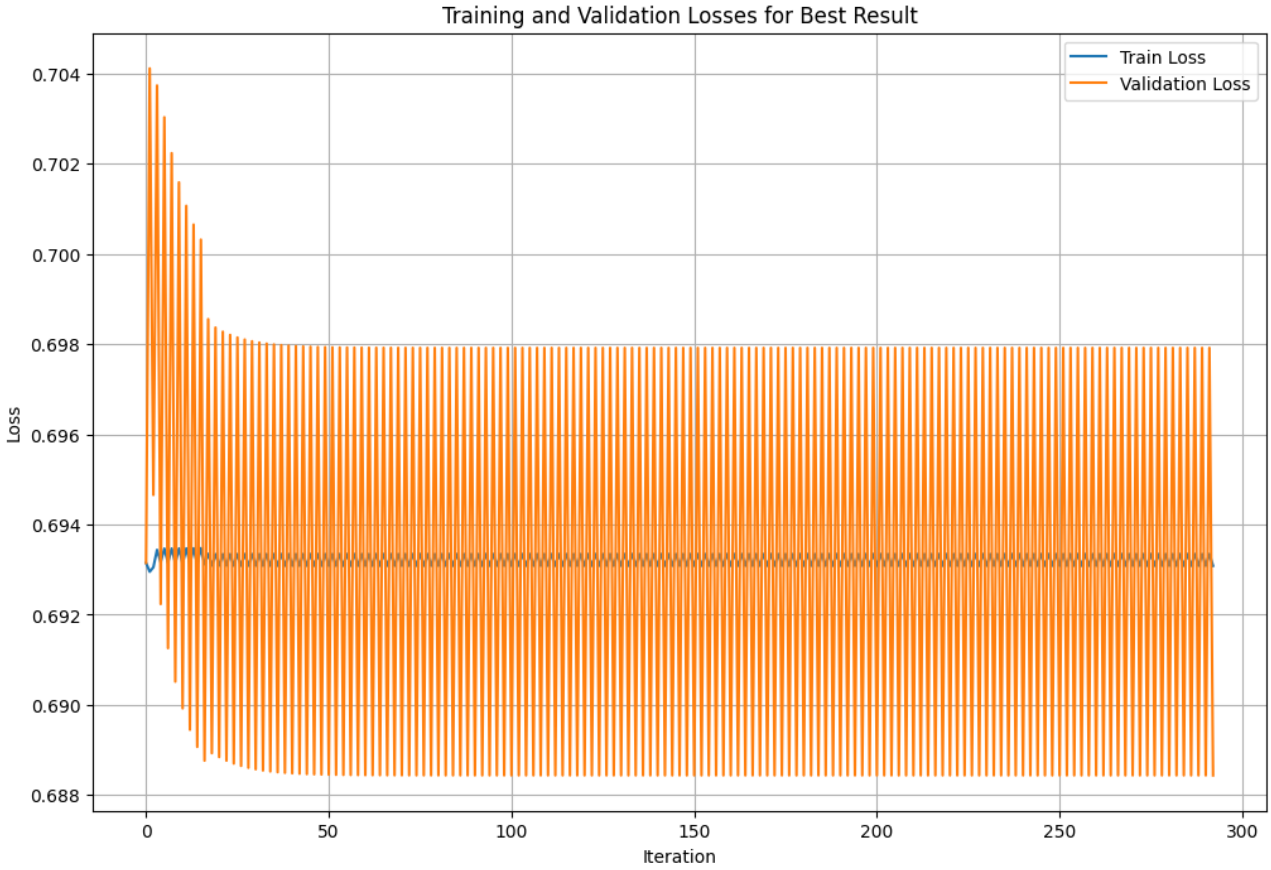
F)

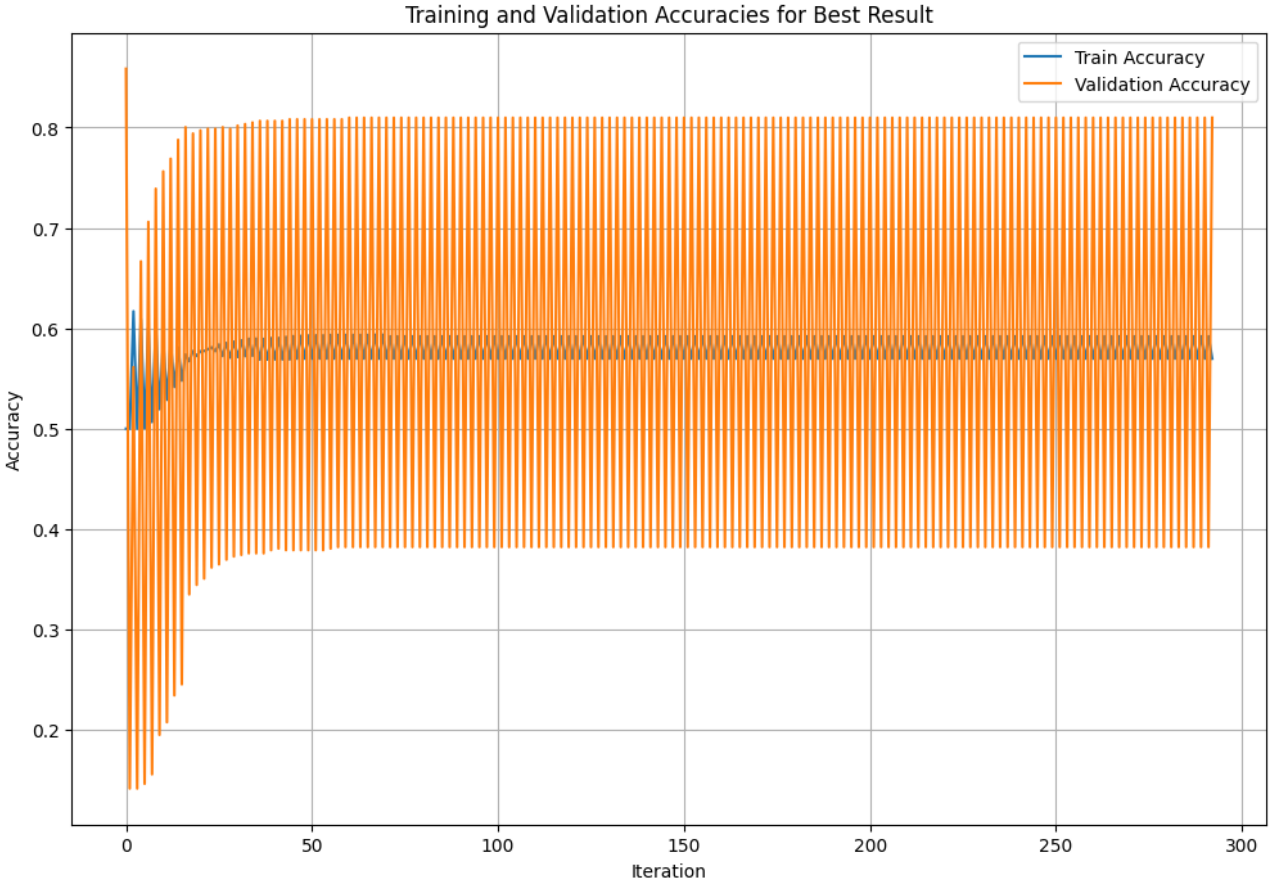
Checked the performance with the following values :

learning_rates = [0.001, 0.05, 0.01, 0.5, 0.1, 0.5, 1]
l1 = [0.001, 0.05, 0.01, 0.5, 0.1, 0.5, 1]
l2 = [0.001, 0.05, 0.01, 0.5, 0.1, 0.5, 1]

Got the best Result with :
Learning Rate: 0.5
L1 Regularization: 0.01
L2 Regularization: 1

Validation Accuracy: 0.8097
Validation Precision: 0.3551
Validation Recall: 0.4222
Validation F1 Score: 0.3858

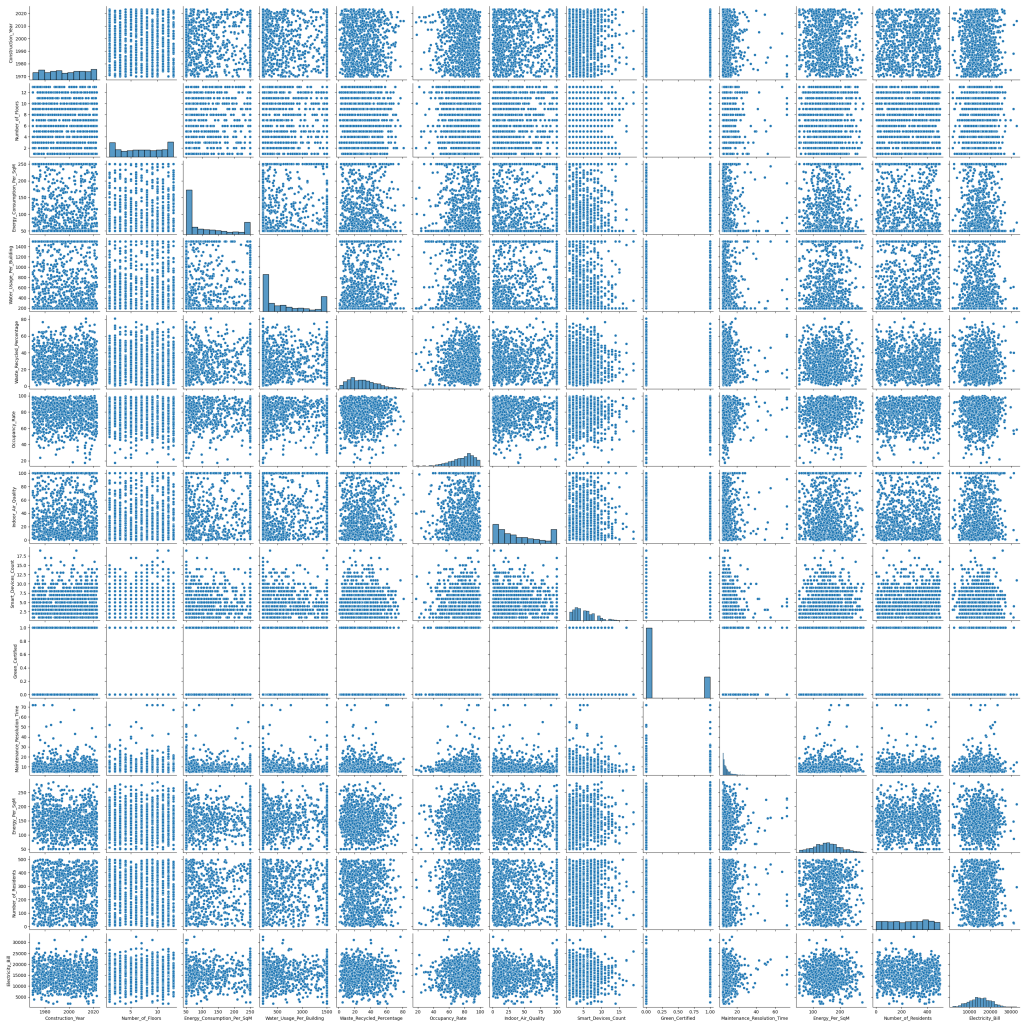

Training and Validation Losses for Best Result
Train Loss
Validation Loss
Loss
Iteration
0.704
0.702
0.700
0.698
0.696
0.694
0.692
0.690
0.688
0
50
100
150
200
250
300


Training and Validation Accuracies for Best Result
Train Accuracy
Validation Accuracy
Accuracy
Iteration
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
50
100
150
200
250
300

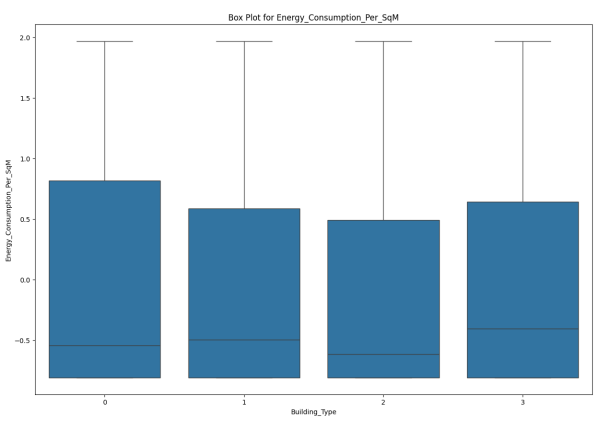
Without early stopping, a model may overfit by learning noise and specific patterns in the training data, leading to poor generalization to unseen data.
With early stopping, the model maintains a balance between fitting the training data and generalizing to new data, resulting in better performance on the validation set.
The provided metrics (validation accuracy: 0.8097, precision: 0.3551, recall: 0.4222, F1 score: 0.3858) indicate a reasonable balance between precision and recall and good generalization as compared to previous results. Therefore Early stopping helps ensure the model performs well on unseen data.

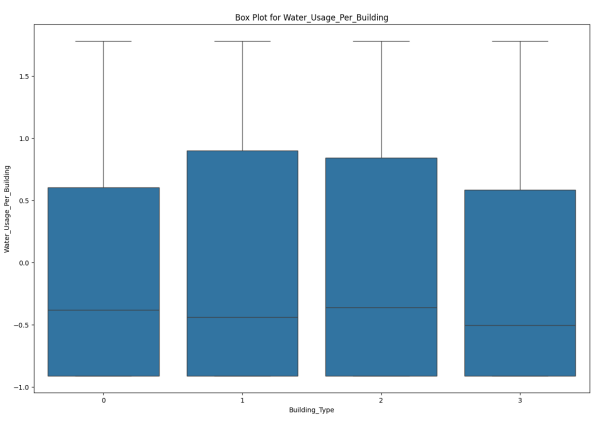
Section C

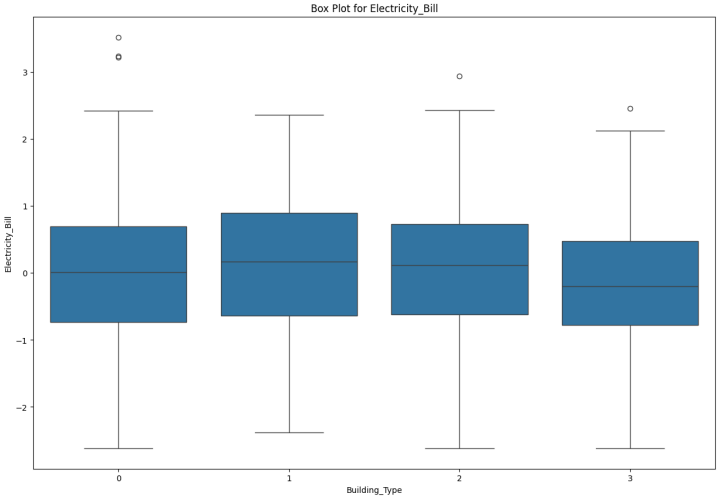
A) Box Plot



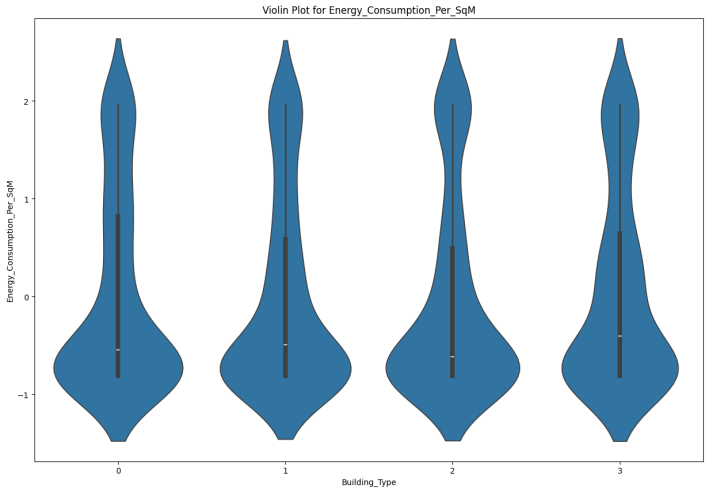
Box Plots

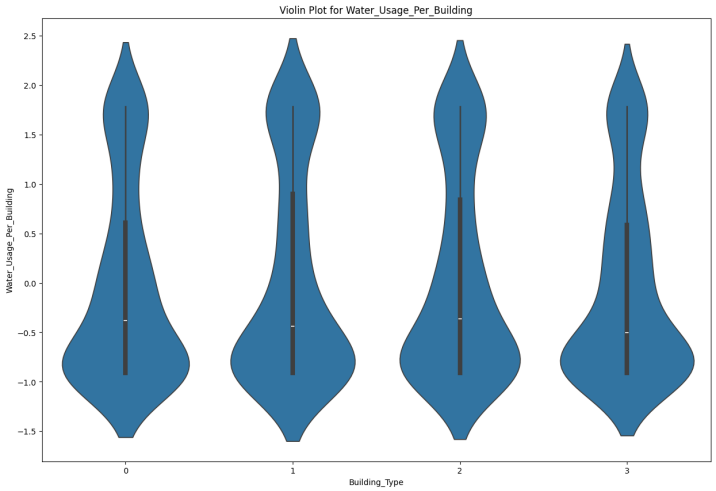


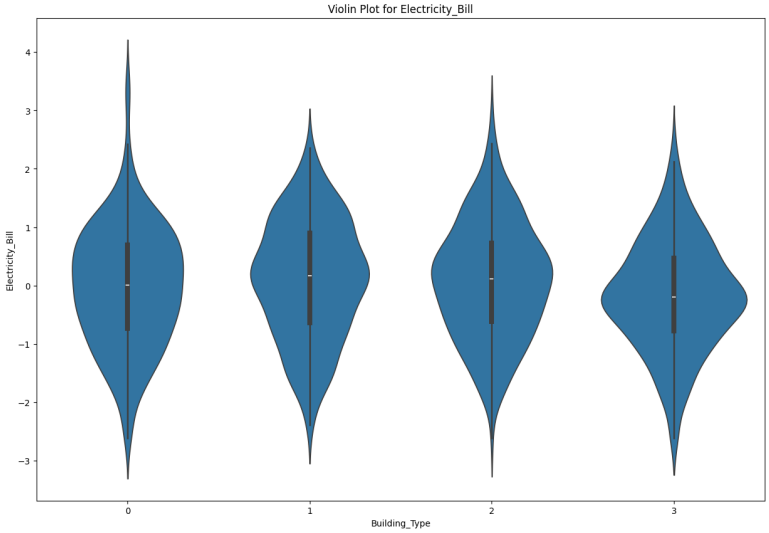




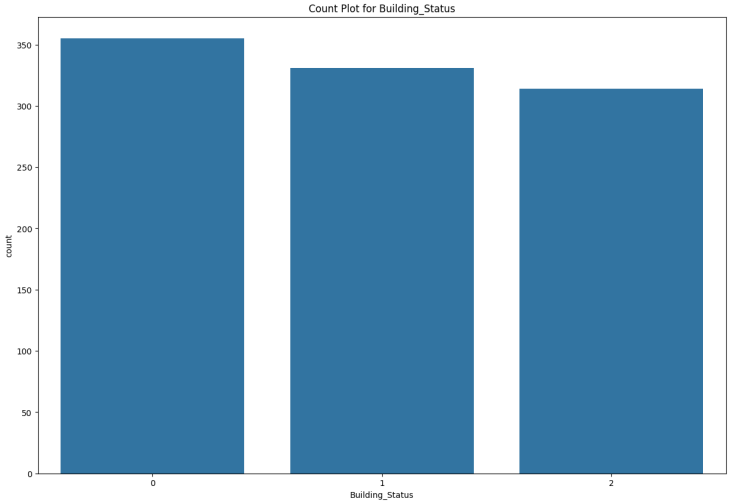
Violin Plots

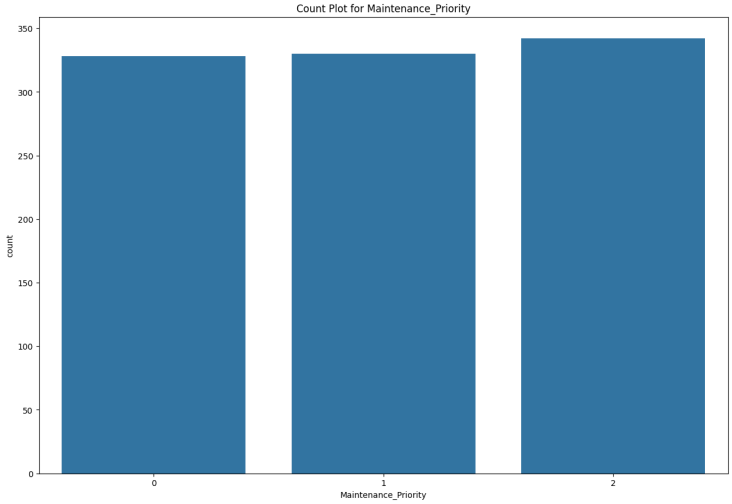




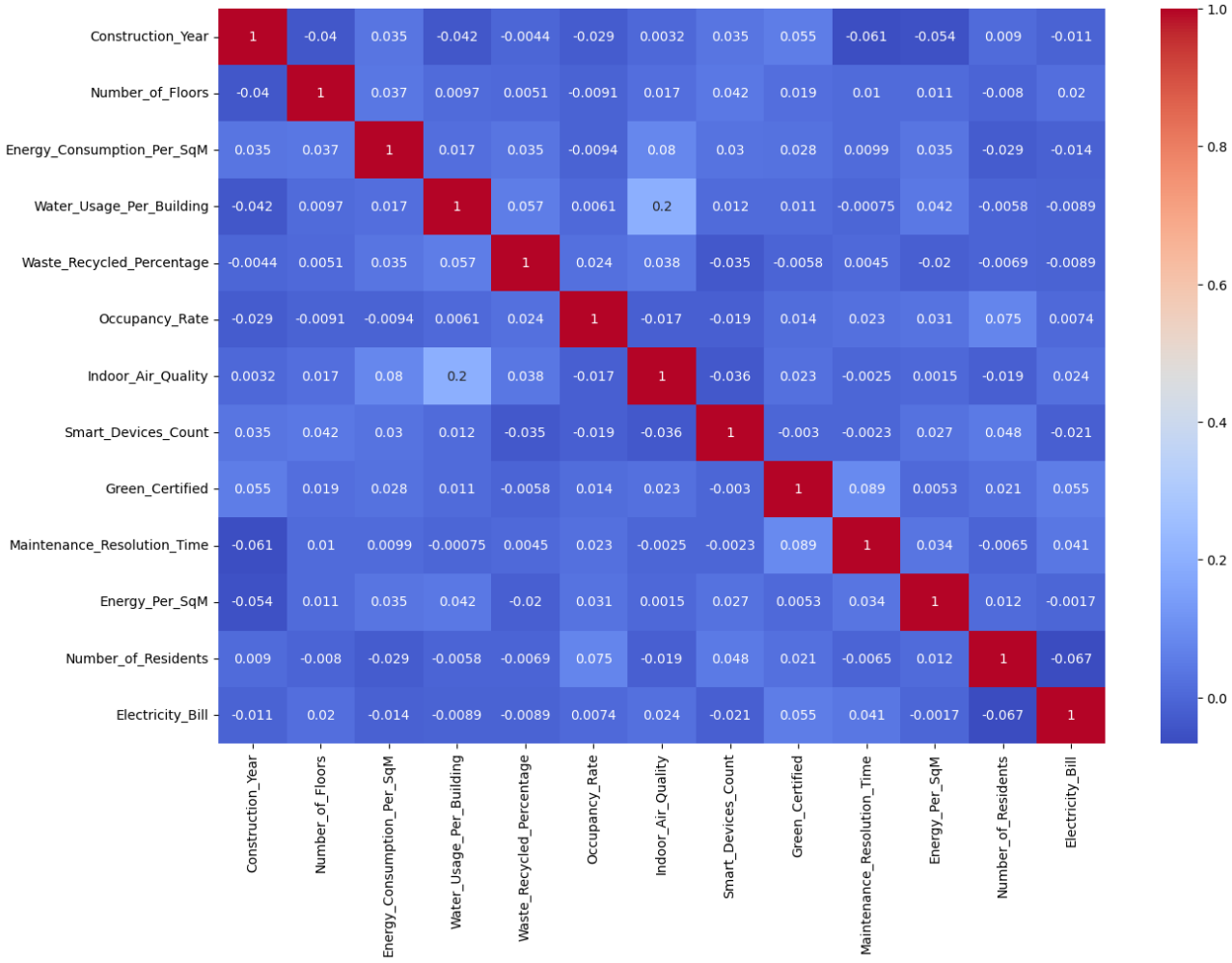


Count Plots





Correlation Heatmap



1) The count plot shows that the most common building status is "Closed," followed by "Operational" and "Under Maintenance."

2) Commercial buildings tend to have higher energy consumption per square meter compared to Residential and Institutional buildings.

3) Institutional buildings show a higher range of water usage, with some buildings using up to 1500 units of water.

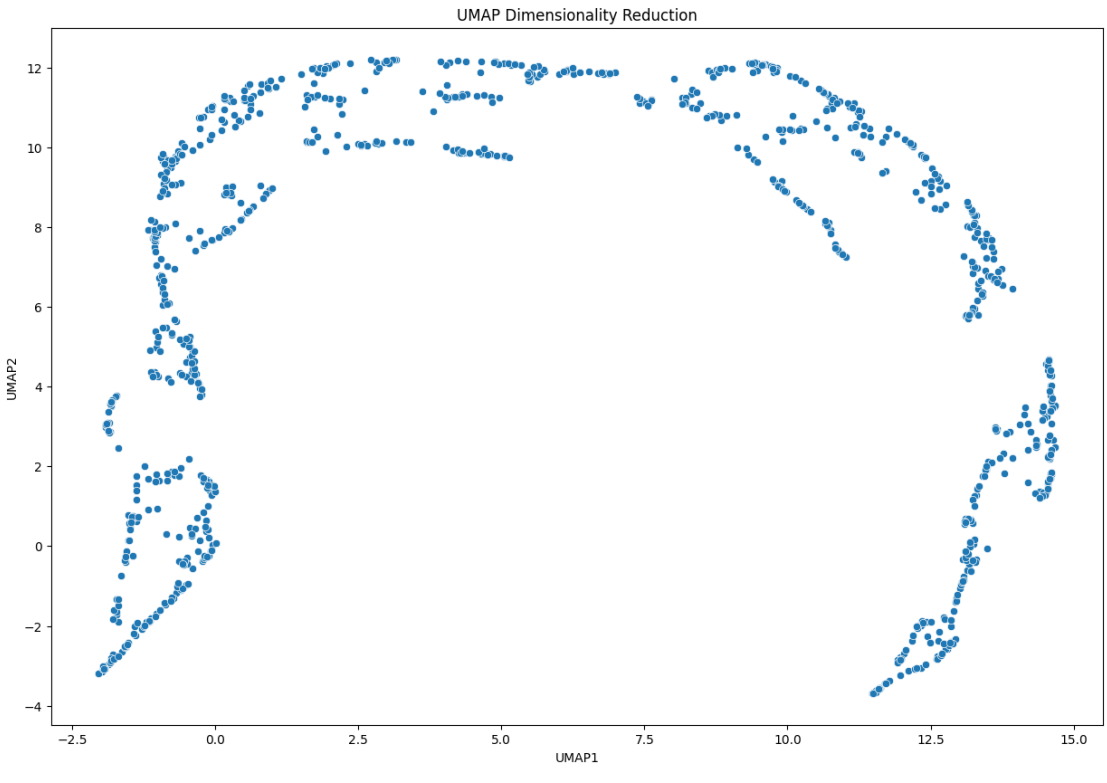
4) The count plot for maintenance priority indicates that the dataset has a balanced distribution of Low, Medium, and High priority maintenance cases.

5) Construction Year: The construction year of buildings does not show a strong correlation with other numerical features, indicating that the age of the building might not be a significant factor in energy or water usage.

6) Number of Floors: The number of floors in a building shows a weak correlation with energy consumption and water usage, suggesting that taller buildings do not necessarily consume more energy or water per square meter.

7) Electricity Bill: The electricity bill does not show a strong correlation with other numerical features, indicating that factors other than those measured might be influencing the electricity costs.

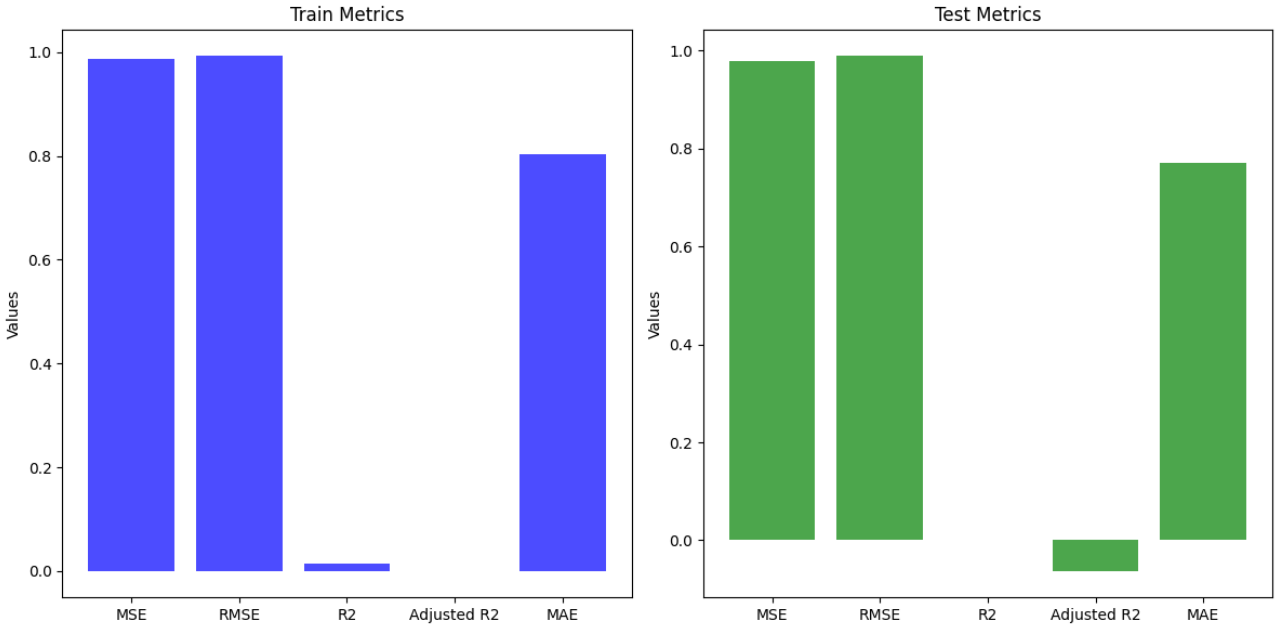
B)



- The UMAP plot shows distinct clusters, indicating that the data points are grouped together based on similarities in the high-dimensional space. Some clusters are more densely packed, suggesting stronger similarities among those data points.

- There are clear separations between some clusters, indicating that UMAP has effectively reduced the dimensionality while preserving the structure of the data. However, there are also areas where clusters are closer together, which might indicate overlapping features or less distinct separability in those regions.

C)



Train Metrics:
MSE: 0.9860774791553897
RMSE: 0.9930143398538561
R2: 0.013922520844610431
Adjusted R2: -0.0011091480449534341
MAE: 0.8041565546016305

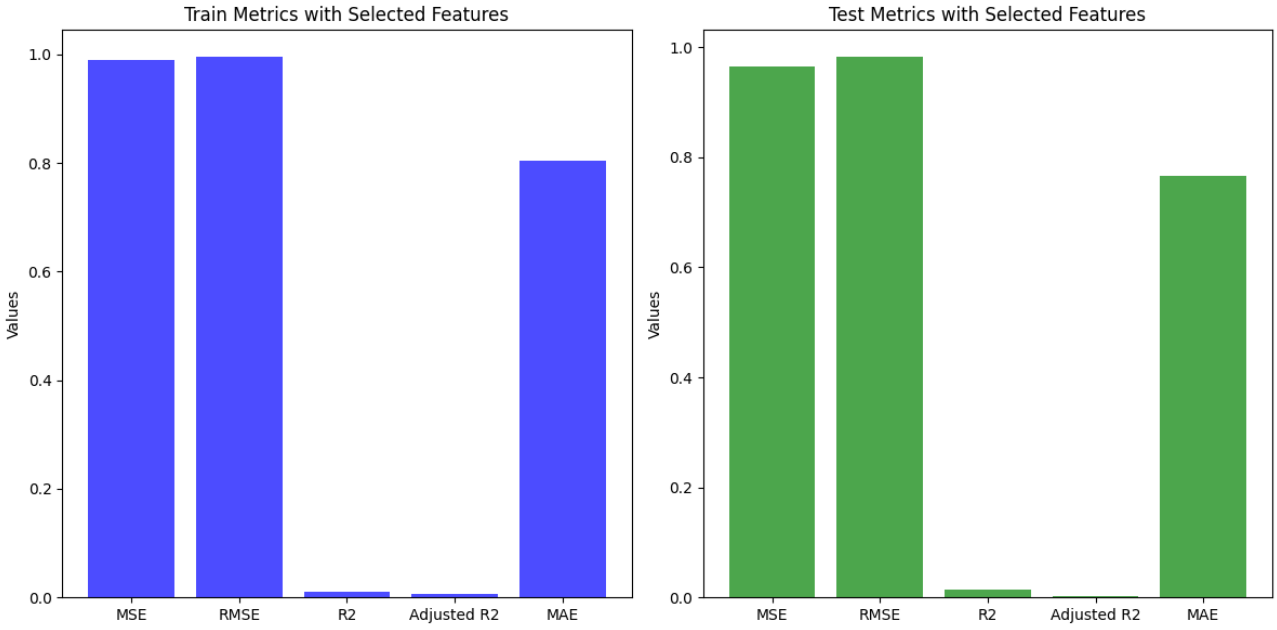
Test Metrics:
MSE: 0.9781406369408592
RMSE: 0.989009927625026
R2: 3.7344733075483916e-05
Adjusted R2: -0.0640628254763429
MAE: 0.7712544435163373

- The low R2 and adjusted R2 scores, along with high MSE, RMSE, and MAE values, indicate that the model is not capturing the underlying patterns in the data.
- The similar performance on both training and testing datasets suggests that the model is not overfitting but rather underfitting. This means the model is too simple to capture the complexity of the data.

D) Train Metrics with Selected Features:

MSE: 0.9898654545087161
RMSE: 0.9949198231559747
R2: 0.010134545491284008
Adjusted R2: 0.007153023037944517
MAE: 0.8041856408006224

Test Metrics with Selected Features:
MSE: 0.9645790232577828
RMSE: 0.9821298403254952
R2: 0.013901513867941251
Adjusted R2: 0.0018759225736478813
MAE: 0.7655416698015712



Train Metrics:

MSE and RMSE: Both metrics are similar compared to using all features.
R2 and Adjusted R2: The R2 score and adjusted R2 score are slightly lower with the selected features, indicating a minor decrease in the model's ability to explain the variance in the training data.
MAE: The MAE is almost the same in both cases.

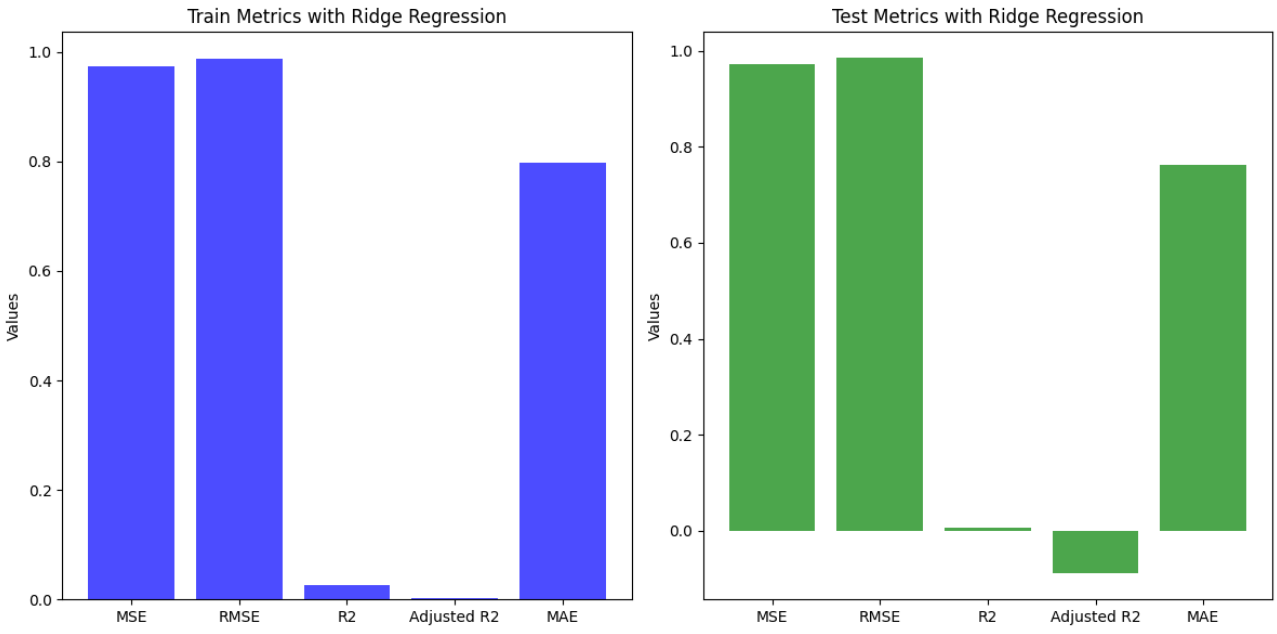
Test Metrics:

MSE and RMSE: Both metrics are slightly better with the selected features compared to using all features.
R2 and Adjusted R2: The R2 score and adjusted R2 score are slightly higher with the selected features, indicating a minor improvement in the model's ability to explain the variance in the test data.
MAE: The MAE is slightly better with the selected features.

E)

Train Metrics with Ridge Regression:
MSE: 0.974551476915771
RMSE: 0.9871937382883722
R2: 0.02544852308422907
Adjusted R2: 0.003503658711509594
MAE: 0.7982132190009789

Test Metrics with Ridge Regression:
MSE: 0.9721406419770157
RMSE: 0.9859719275806059
R2: 0.006171197748729207
Adjusted R2: -0.09014701216106545
MAE: 0.7622417751111837



Train Metrics:

MSE and RMSE: Ridge Regression shows a slight improvement in MSE and RMSE compared to both the original model and the model with selected features.
R2 and Adjusted R2: Ridge Regression has a slightly higher R2 score compared to the original model and the model with selected features, indicating a better fit on the training data.
MAE: Ridge Regression has a slightly lower MAE compared to the original model and the model with selected features.

Test Metrics:
MSE and RMSE: Ridge Regression shows a slight improvement in MSE and RMSE compared to the original model and the model with selected features.

R2 and Adjusted R2: Ridge Regression has a slightly higher R2 score compared to the original model but lower than the model with selected features. The adjusted R2 score is negative, indicating that the model is not performing well on the test data.
MAE: Ridge Regression has a slightly lower MAE compared to the original model and the model with selected features.

Ridge Regression with One-Hot Encoding shows a slight improvement in most metrics compared to the original model and the model with selected features. However, the improvement is not substantial.
Underfitting: The similar performance on both training and testing datasets suggests that the model is still underfitting. This means the model is too simple to capture the complexity of the data.

F)
Train Metrics with ICA (4 components):
MSE: 0.9990214640526406
RMSE: 0.9995106122761481
R2: 0.0009785359473595268
Adjusted R2: -0.0030376307422994575
MAE: 0.8047634392568596

Test Metrics with ICA (4 components):
MSE: 0.9809877504971333
RMSE: 0.9904482573547865
R2: -0.0028732870555008283
Adjusted R2: -0.01924672847681519
MAE: 0.7711420629273451

Train Metrics with ICA (5 components):
MSE: 0.9928977052035752
RMSE: 0.9964425247868415
R2: 0.0071022947964248795
Adjusted R2: 0.0021078395388616222
MAE: 0.806083231908769

Test Metrics with ICA (5 components):
MSE: 0.98780892805077
RMSE: 0.9938857721341875
R2: -0.009846642993266874
Adjusted R2: -0.03054022174312898
MAE: 0.7738287184522998

Train Metrics with ICA (6 components):
MSE: 0.9928014193911952
RMSE: 0.996394208830619
R2: 0.007198580608804961
Adjusted R2: 0.0011997804916376031
MAE: 0.8061518875349377

Test Metrics with ICA (6 components):
MSE: 0.9860827224421028
RMSE: 0.9930169799364474
R2: -0.008081926265641526
Adjusted R2: -0.032972838025286944
MAE: 0.7728300223223185

Train Metrics with ICA (8 components):
MSE: 0.99222762695601
RMSE: 0.9961062327663701
R2: 0.007772373043990144
Adjusted R2: -0.00023753716352548082
MAE: 0.8072096793569769

Test Metrics with ICA (8 components):
MSE: 0.9892178099039143
RMSE: 0.994594294123948
R2: -0.011286956569475537
Adjusted R2: -0.044856648073856586
MAE: 0.775287665450525

The use of ICA with Ridge Regression does not show significant improvement over the original model using all features.

G)

```
Train Metrics with ElasticNet (alpha=0.01):
MSE: 0.9756660931025176
RMSE: 0.9877581146730801
R2: 0.024333906897482493
Adjusted R2: 0.002363943695583459
MAE: 0.7979739721500393

Test Metrics with ElasticNet (alpha=0.01):
MSE: 0.9664132197664127
RMSE: 0.9830631819809003
R2: 0.012026396996417388
Adjusted R2: -0.08372434866912815
MAE: 0.7602923893067804

Train Metrics with ElasticNet (alpha=0.05):
MSE: 0.987582604735171
RMSE: 0.9937719078013681
R2: 0.0124173952648291
Adjusted R2: -0.009820902897068384
MAE: 0.8015713755890869

Test Metrics with ElasticNet (alpha=0.05):
MSE: 0.9643043173221105
RMSE: 0.9819899782187752
R2: 0.014182348408985912
Adjusted R2: -0.08135945042362347
MAE: 0.7628612334189931

Train Metrics with ElasticNet (alpha=0.1):
MSE: 0.9975707958729265
RMSE: 0.9987846594100885
R2: 0.002429204127073592
Adjusted R2: -0.02003400724365756
MAE: 0.8045653461552231

Test Metrics with ElasticNet (alpha=0.1):
MSE: 0.9780223585073368
RMSE: 0.9889501294339047
R2: 0.0001582619225987246
Adjusted R2: -0.09674269947734482
MAE: 0.7696135354466296

Train Metrics with ElasticNet (alpha=1):
MSE: 1.0000000000000002
RMSE: 1.0
R2: 0.0
Adjusted R2: -0.022517911975435068
MAE: 0.8049719862860575
```

Test Metrics with ElasticNet (alpha=1):
MSE: 0.9816940017294083
RMSE: 0.9908047243172634
R2: -0.0035952945366743982
Adjusted R2: -0.10086003673846666
MAE: 0.7712446669107382

Train Metrics with ElasticNet (alpha=5):
MSE: 1.0000000000000002
RMSE: 1.0
R2: 0.0
Adjusted R2: -0.022517911975435068
MAE: 0.8049719862860575

Test Metrics with ElasticNet (alpha=5):
MSE: 0.9816940017294083
RMSE: 0.9908047243172634
R2: -0.0035952945366743982
Adjusted R2: -0.10086003673846666
MAE: 0.7712446669107382

MSE: The MSE is lowest for ElasticNet with alpha=0.05, indicating better performance compared to other alpha values.

RMSE: The RMSE follows the same trend as MSE, with ElasticNet (alpha=0.05) performing slightly better.

R2: The R2 score is highest for ElasticNet (alpha=0.05), indicating a slightly better fit compared to the other models.

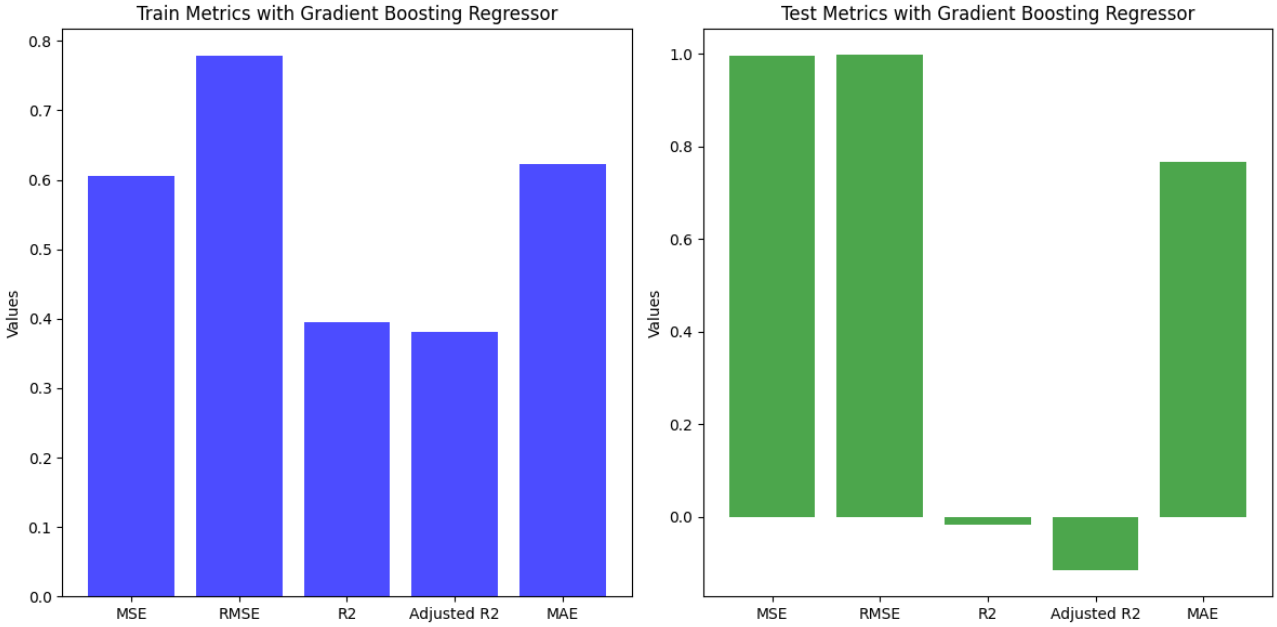
Adjusted R2: The Adjusted R2 score is also highest for ElasticNet (alpha=0.05), though all models have negative values indicating poor fit.

MAE: The MAE is lowest for ElasticNet (alpha=0.01), suggesting it has the smallest average error.

Overall, ElasticNet with alpha=0.05 performs slightly better than the other models on the test dataset in terms of MSE, RMSE, R2, and Adjusted R2. However, ElasticNet with alpha=0.01 has the lowest MAE.

H)
Train Metrics with Gradient Boosting Regressor:
MSE: 0.6056001988780503
RMSE: 0.7782031861140446
R2: 0.39439980112194983
Adjusted R2: 0.38076294915130804
MAE: 0.6227447613203396

Test Metrics with Gradient Boosting Regressor:
MSE: 0.9948394442741598
RMSE: 0.9974163846028197
R2: -0.017034007882456237
Adjusted R2: -0.11560118045256207
MAE: 0.7664125705799628



Analysis Train Metrics:
MSE and RMSE: The Gradient Boosting Regressor shows a significant improvement in MSE and RMSE compared to both the original model and the ElasticNet model.
R2 and Adjusted R2: The R2 and adjusted R2 scores are much higher with the Gradient Boosting Regressor, indicating a better fit on the training data.
MAE: The MAE is lower with the Gradient Boosting Regressor, indicating better performance on the training data.

Test Metrics:
MSE and RMSE: The MSE and RMSE values are slightly worse with the Gradient Boosting Regressor compared to the original model and the ElasticNet model.

R2 and Adjusted R2: The R2 and adjusted R2 scores are lower with the Gradient Boosting Regressor, indicating a decrease in the model's ability to explain the variance in the test data.
MAE: The MAE is slightly better with the Gradient Boosting Regressor compared to the original model and the ElasticNet model.

The Gradient Boosting Regressor shows a significant improvement in the training metrics, indicating that it captures the underlying patterns in the training data better than the previous models.

Slight Decrease on Test Data: The performance on the test data is slightly worse compared to the original model and the ElasticNet model, indicating that the model may not generalize as well to unseen data.