ML Assign 1

Date @September 13, 2024

Section A (Theoretical)

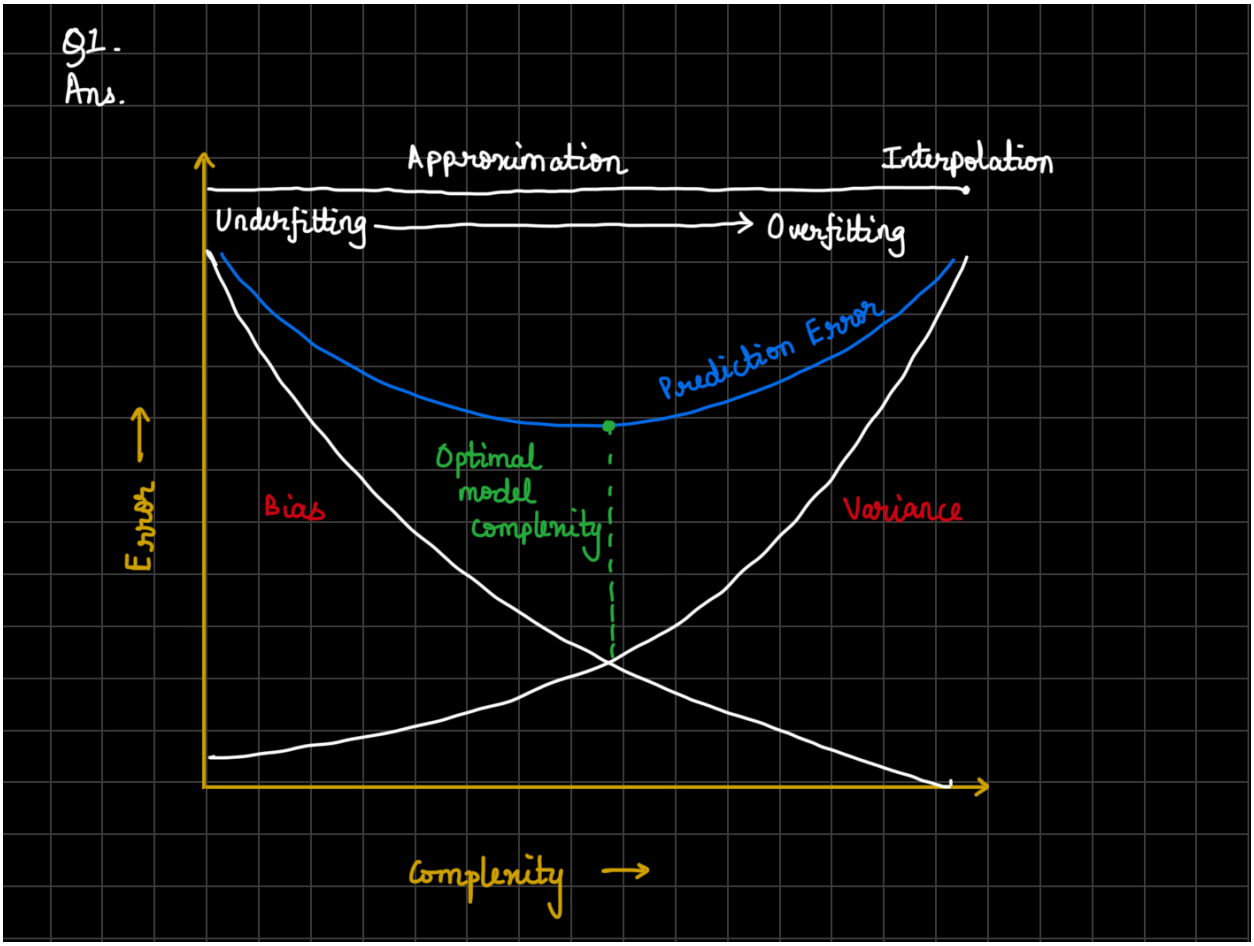
Ques a. You are developing a machine-learning model for a prediction task. As you increase the complexity of your model, for example, by adding more features or by including higher-order polynomial terms in a regression model, what is most likely to occur? Explain in terms of bias and variance with suitable graphs as applicable.

Ans. Bias: While training, a machine learning model analyses and find patterns in the dataset and uses these patterns in the dataset for prediction. There are likely to be differences in the prediction made by the model and the actual values. And these differences are called **bias**. High bias means that the best fit line is in the form of a straight line. Therefore, not fitting the data accurately. Such a form of fitting is called **under-fitting**.

Variance: The amount of variation in the prediction of the data from the actual prediction when a different dataset has been used which the model has never seen before is called the **variance**. The best fit line for a high variance model fits very complexly to the training data. Thus, the variation is high when the best fit line tries to fit the data which the model has never seen before, giving inaccurate predictions. Such a form of fitting is called o**ver-fitting**.

As the complexity of the model increases:

- **Bias decreases** because the model better fits the training data.
- **Variance increases** because the model becomes more sensitive to small fluctuations in the training data.



Ques b. You’re working at a tech company that has developed an advanced email filtering system to ensure users’ inboxes are free from spam while safeguarding legitimate messages. After the model has been trained, you are tasked with evaluating its

performance on a validation dataset containing a mix of spam and legitimate emails. The results show that the model successfully identified 200 spam emails. However, 50 spam emails managed to slip through, being incorrectly classified as legitimate. Meanwhile, the system correctly recognised most of the legitimate emails, with 730 reaching the users’ inboxes as intended. Unfortunately, the filter mistakenly flagged 20 legitimate emails as spam, wrongly diverting them to the spam folder. You are asked to assess the model by calculating an average of its overall classification performance across the different categories of emails.

Ans.

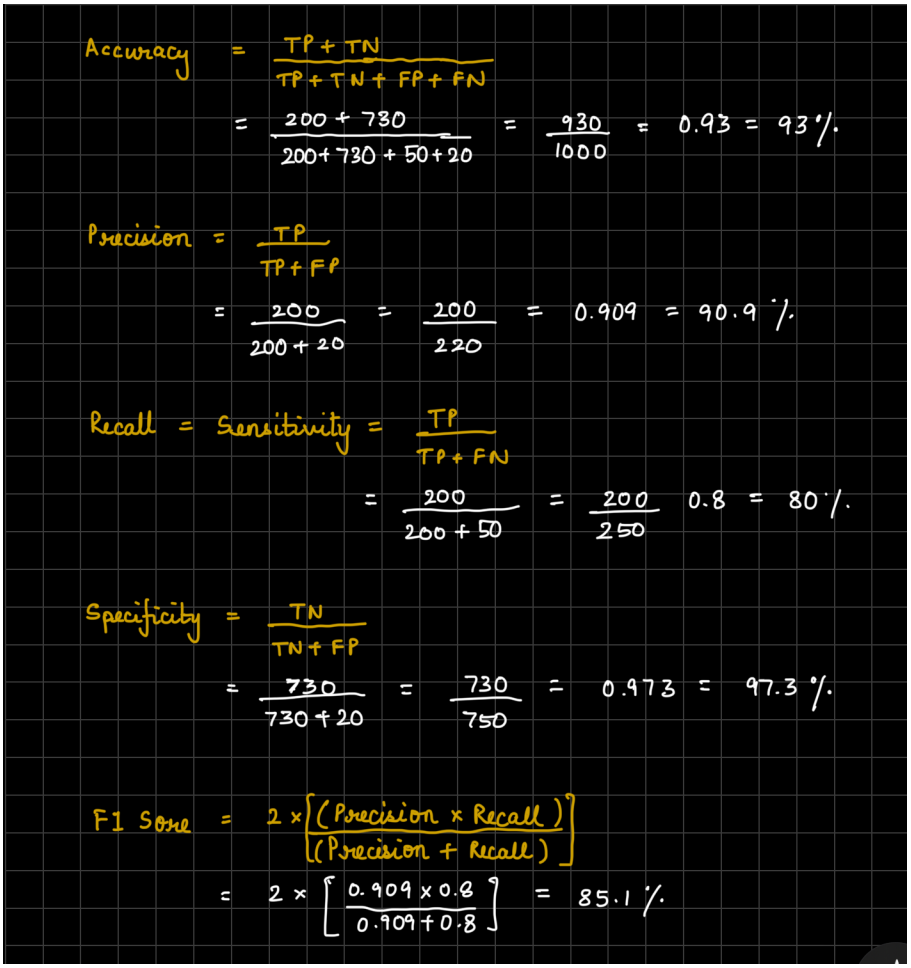
Actual Class	Predicted Class	Category
Spam	Spam	TP = 200
Spam	Legitimate	FN = 50
Legitimate	Legitimate	TN = 730
Legitimate	Spam	FP = 20

True Positive(TP): Spam emails correctly classified as spam.

False Negative(FN): Spam emails incorrectly classified as legitimate.

True Negative(TN): Legitimate emails correctly classified as legitimate.

False Positive(FP): Legitimate emails incorrectly classified as spam.



So, now let’s assume that detecting if a mail is legitimate or not is our positive class.

This means that,

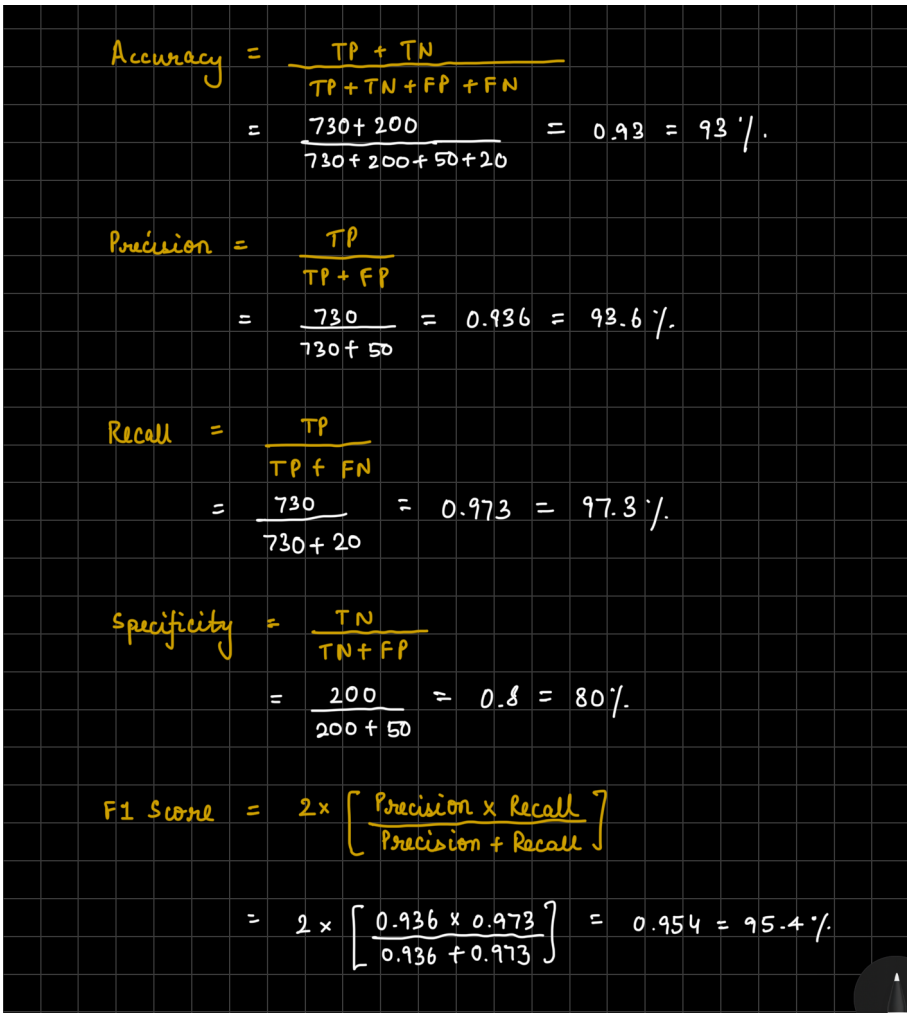
Actual Class	Predicted Class	Category
Legitimate	Legitimate	TP = 730
Legitimate	Spam	FN = 20
Spam	Spam	TN = 200
Spam	Legitimate	FP = 50

True Positive(TP): Legitimate emails correctly classified as legitimate.

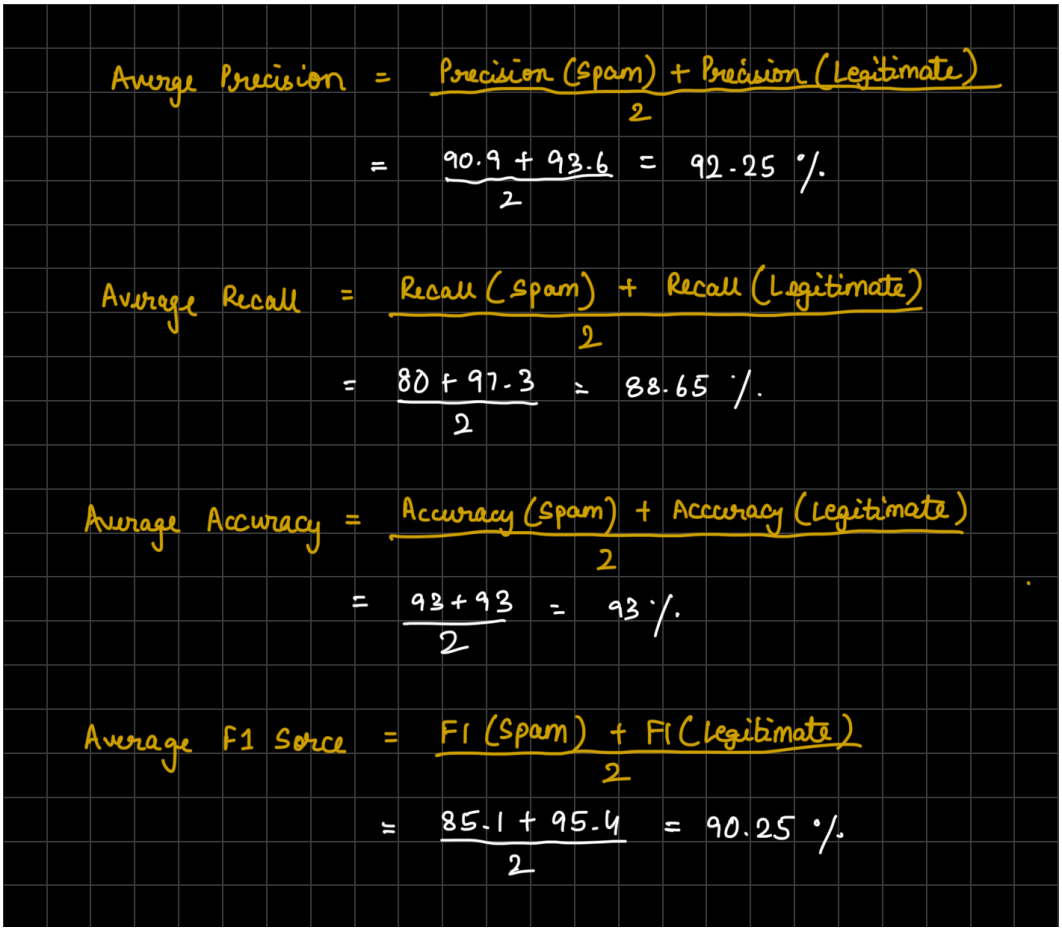
False Negative(FN): Legitimate emails incorrectly classified as spam.

True Negative(TN): Spam emails correctly classified as spam.

False Positive(FP): Spam emails incorrectly classified as Legitimate.



And now to calculate the average of the overall classification, we will take average of both spam and legitimate classes.



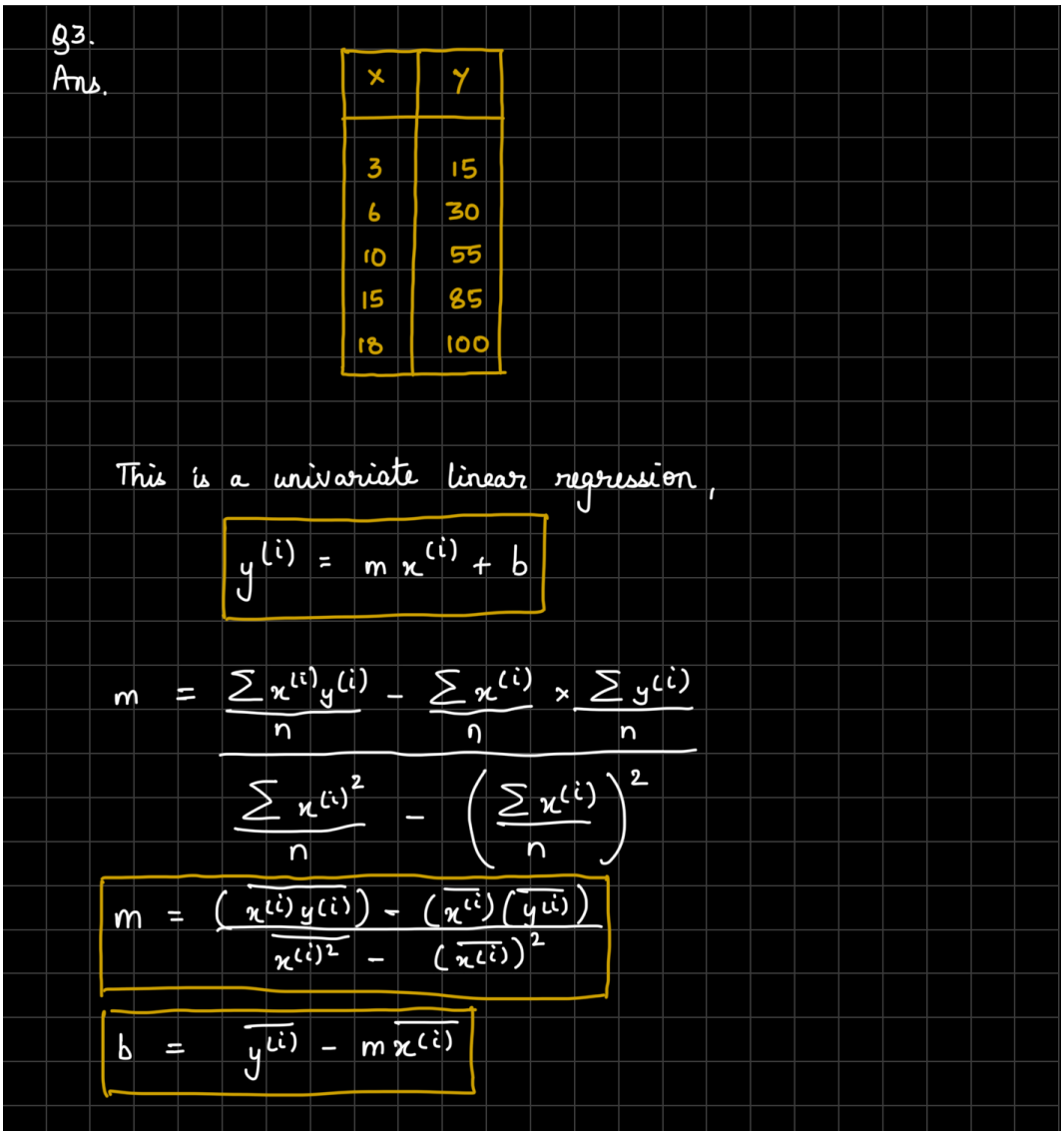
The average F1 score provides a balanced measure of the model's performance across both spam and legitimate classes, considering both precision and recall. Given the results, the F1 score is slightly lower than accuracy, which suggests that while the model performs well overall, there might be a trade-off between precision and recall for different classes.

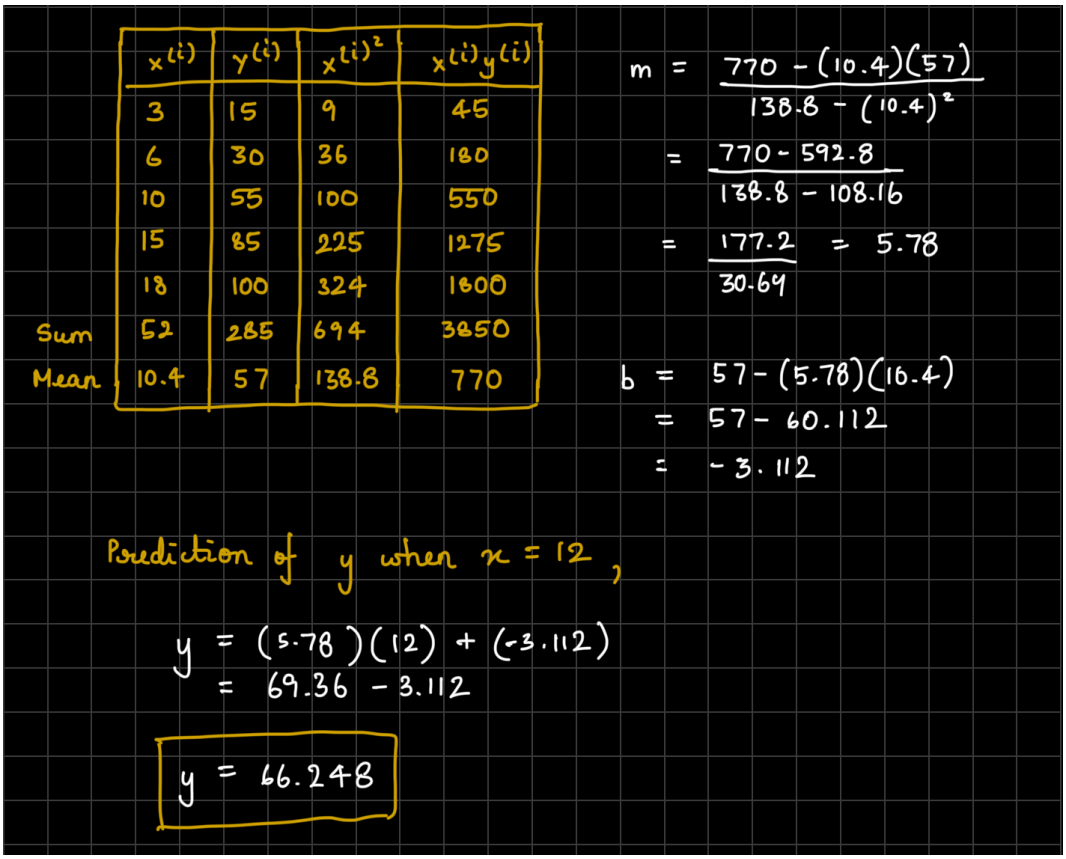
Ques c. Consider the following data where y(units) is related to x(units) over a period of time: Find the equation of the regression line and, using the regression equation obtained, predict the value of y when x=12.

x	y
3	15

6	30
10	55
15	85
18	100

Ans.





Ques d. Given a training dataset with features X and labels Y , let ˆf(X) be the prediction of a model f and L(ˆf(X), Y) be the loss function. Suppose you have two models, f1 and f2, and the empirical risk for f1 is lower than that for f2. Provide a toy example where model f1 has a lower empirical risk on the training set but may not necessarily generalize better than model f2.

Ans. Training data with ***X*** features and ***Y*** labels. Let, the model be ***f.***

$$prediction = \hat{f}(X)$$

$$loss\ function = L(\hat{f}(X), Y)$$

x	y
-2	2

-1	0.5
0	0
1	0.5
2	2

As it is given that the empirical risk of f_1 is lower than f_2, this means that f_1 is a complex function which fits the training data more accurately than f_2.

f_1 is a high degree polynomial function,

$$f_1(x) = 0.1x^5 - 0.5x^4 + 0.3x^3 - 0.1x^2 + 0.2x - 0.2$$

f_2 is a simple linear regression function,

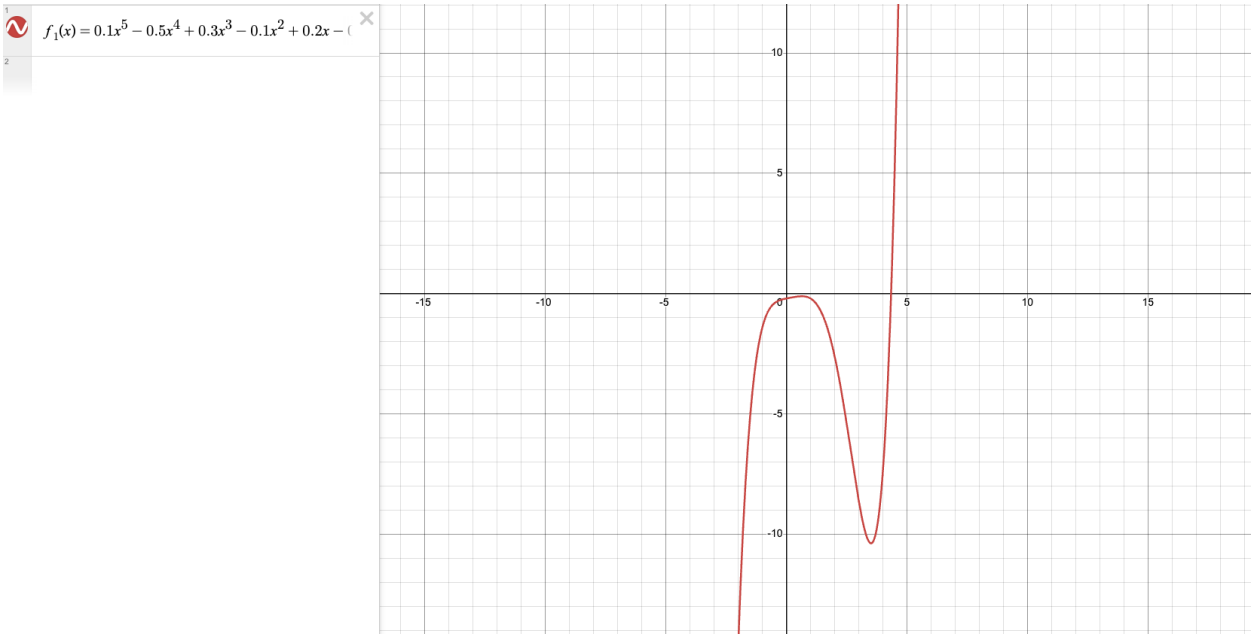
$$f_2(x) = 0.5x$$

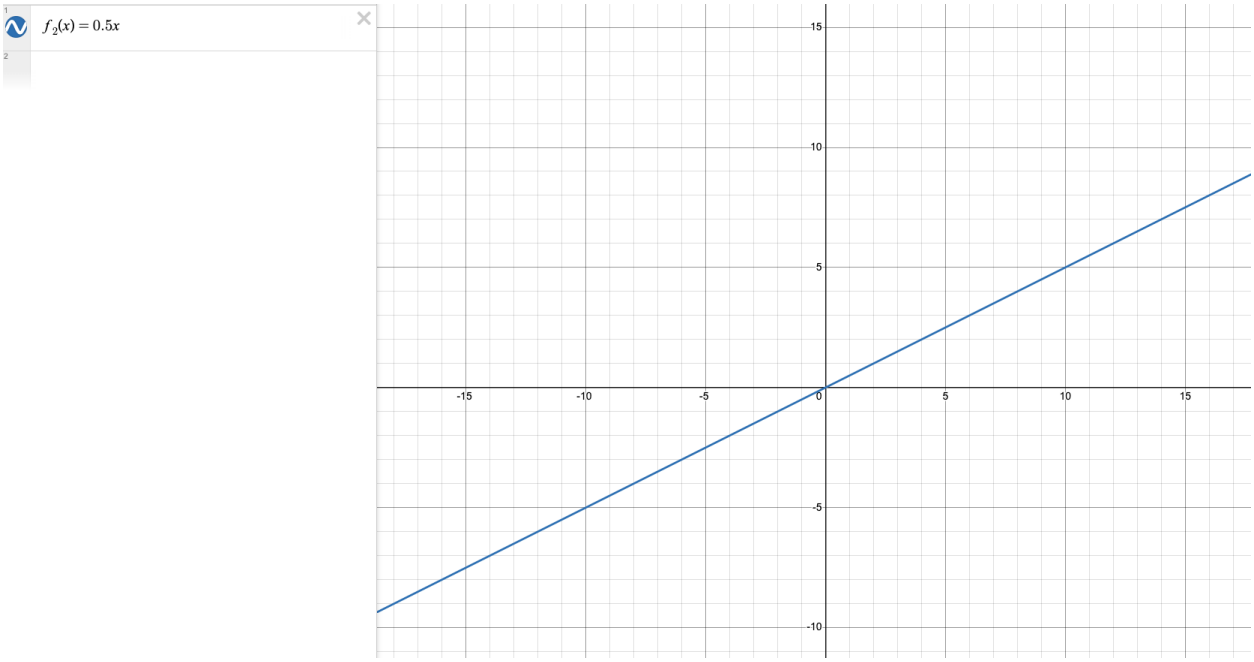
→ Model f_1 will have a **lower empirical risk** on the training data than f_2 because it can fit the training data more accurately. But, when the models are tested on the testing dataset which they have never seen before, model f_1 is more likely to overfit the testing data as it is more specific to the training data. On the other hand, model f_2 will generalise better on the test data inspite of having higher empirical risk on the training data as it captures the broader trend and avoids overfitting.

Bias-Variance tradeoff:

- f_1 has low bias but high variance (overfitting).
- f_2 has higher bias but lower variance (better generalisation).

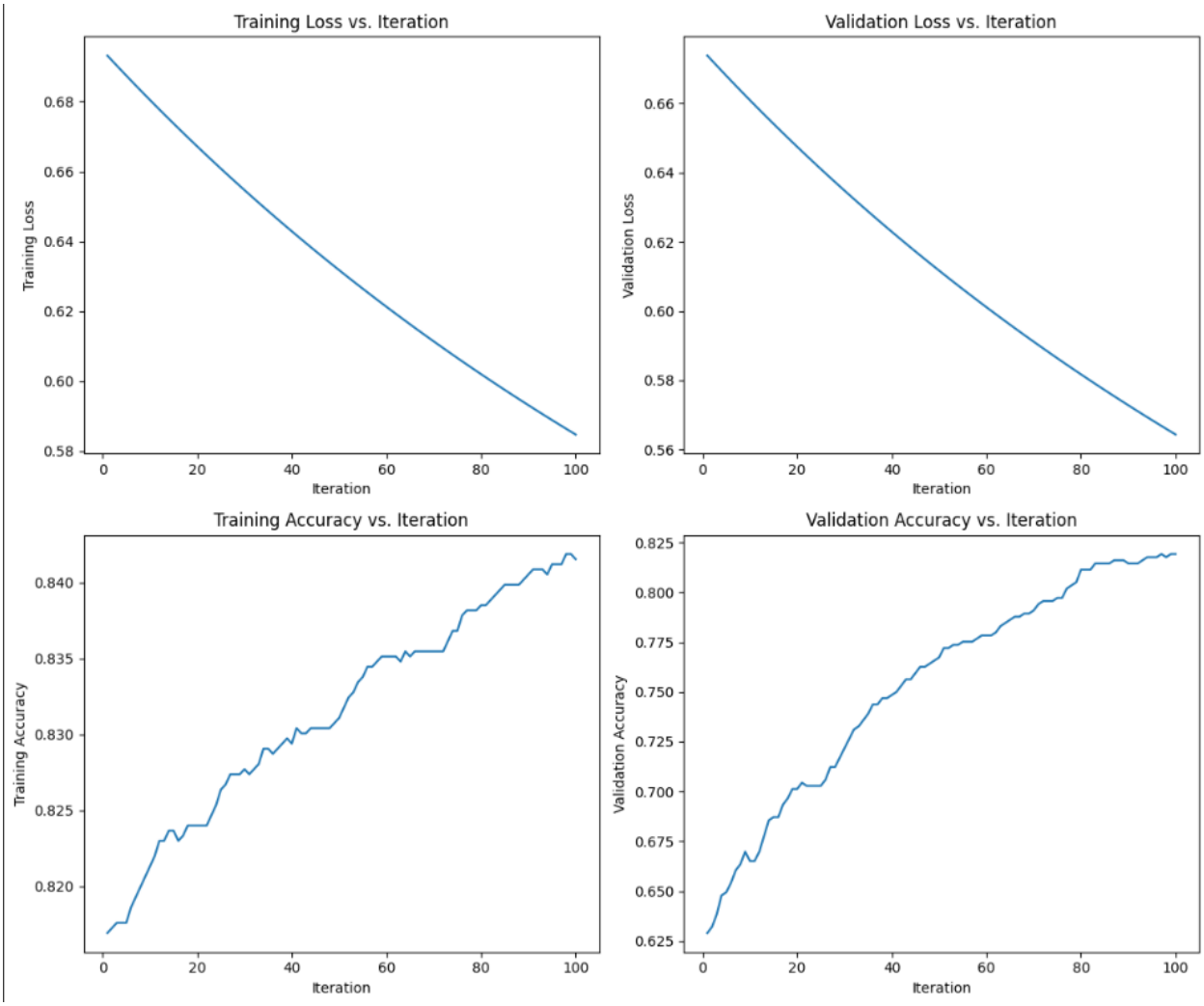
Conclusion: This example demonstrates how a model with a lower training error (like f_1) may not necessarily generalise better than a simpler model (like f_2) that has a higher training error but generalises well on unseen data.





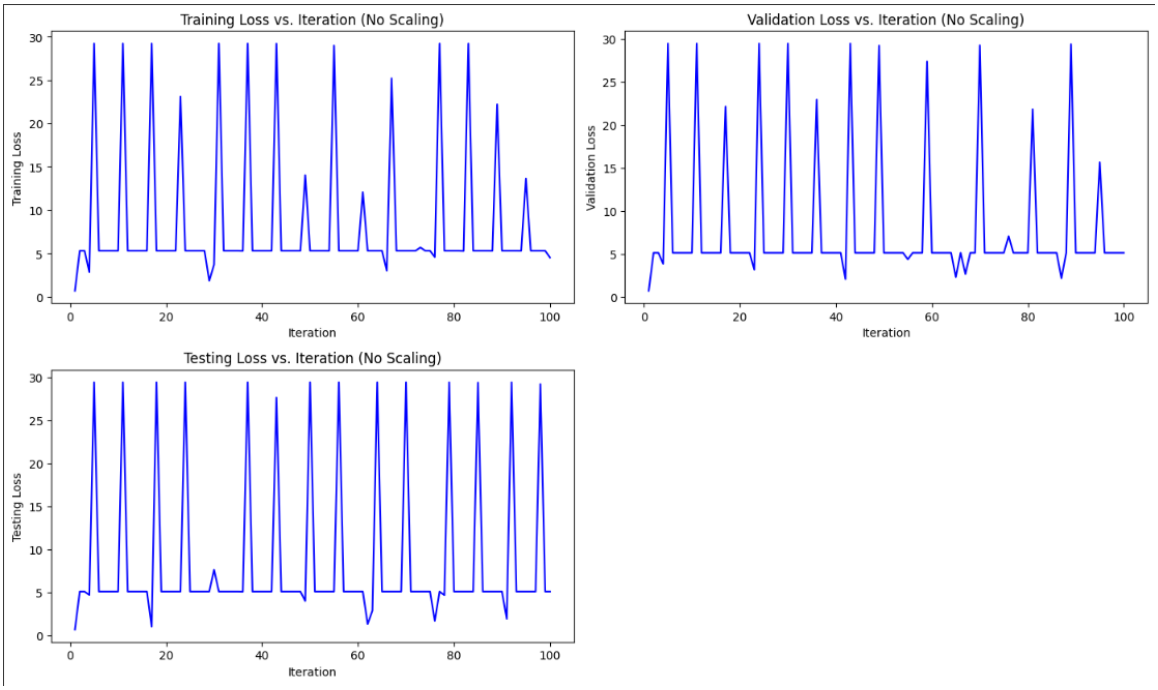
SECTION B

a.

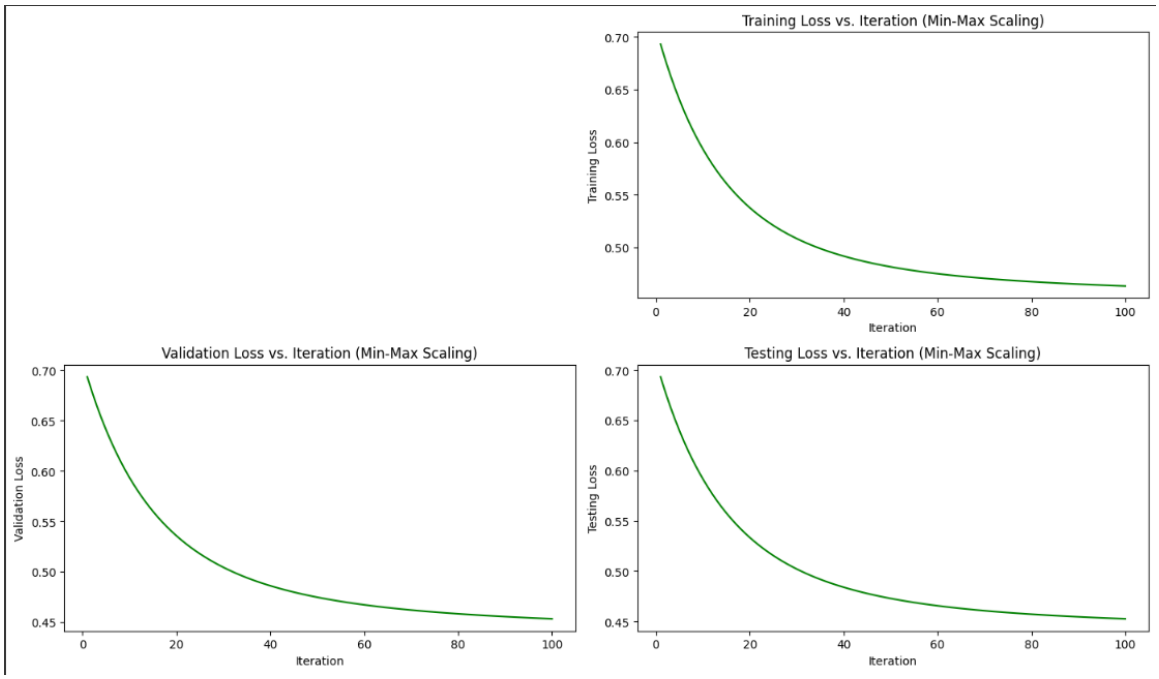


- **Loss:** Both training and validation loss are decreasing, indicating the model is learning.
- **Accuracy:** Both training and validation accuracy are increasing, indicating good performance.
- **Overfitting risk:** Validation loss is starting to plateau, suggesting potential overfitting.

b.



- **Loss:** All three loss curves are fluctuating significantly and not showing a clear decreasing trend. The reason for such plots is because of lack of normalisation and not being able to scale with precision (which comes from libraries).
- **No clear improvement:** The model is not consistently learning, and its performance on both training and validation data is inconsistent.



- **Loss:** All three loss curves are decreasing, indicating that the model is learning. This is because Min-max scaling is helping the model converge faster and more efficiently. By scaling the data, the model can learn more meaningful patterns and generalise better to unseen data.
- **Consistent improvement:** The model's performance is consistently improving on both training, validation, and testing data.
- **No overfitting:** The validation loss is decreasing at a similar rate to the training loss, suggesting that the model is not overfitting.

c. **Metrics:**

```
confustion_matrix = array([[ 29.,  43.],
                                                [ 65., 4
99.]])
```

```
Accuracy: 0.8301886792452831
Precision: 0.4027777777777778
Recall: 0.30851063829787234
f1_score: 0.3493975903614458
```

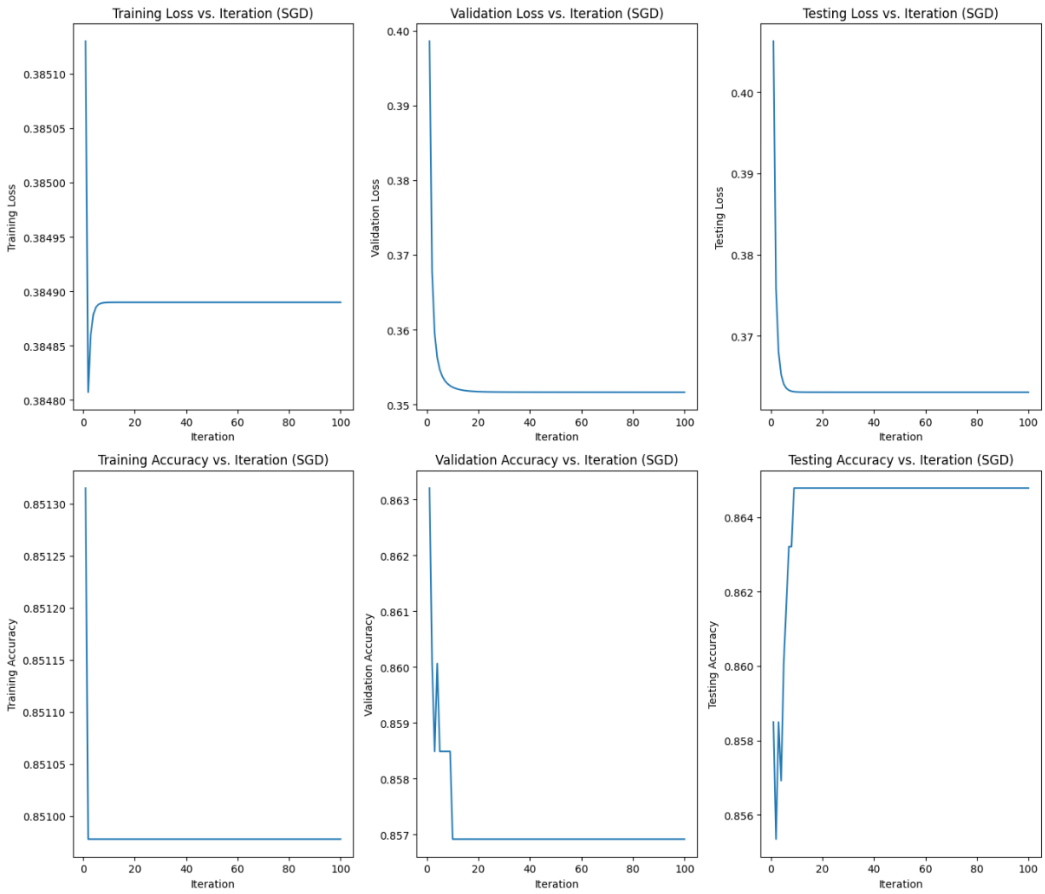
Accuracy: 83% indicates that the model correctly predicted 83% of the samples.

Precision: 40% suggests that when the model predicts a positive class, it's correct only 40% of the time. This indicates low precision, meaning the model might be predicting too many false positives.

Recall: 31% means the model correctly identifies only 31% of the actual positive cases. This indicates low recall, meaning the model might be missing many true positive cases.

F1-score: 35% is a harmonic mean of precision and recall. A low F1-score indicates a need for improvement in both precision and recall.

d.



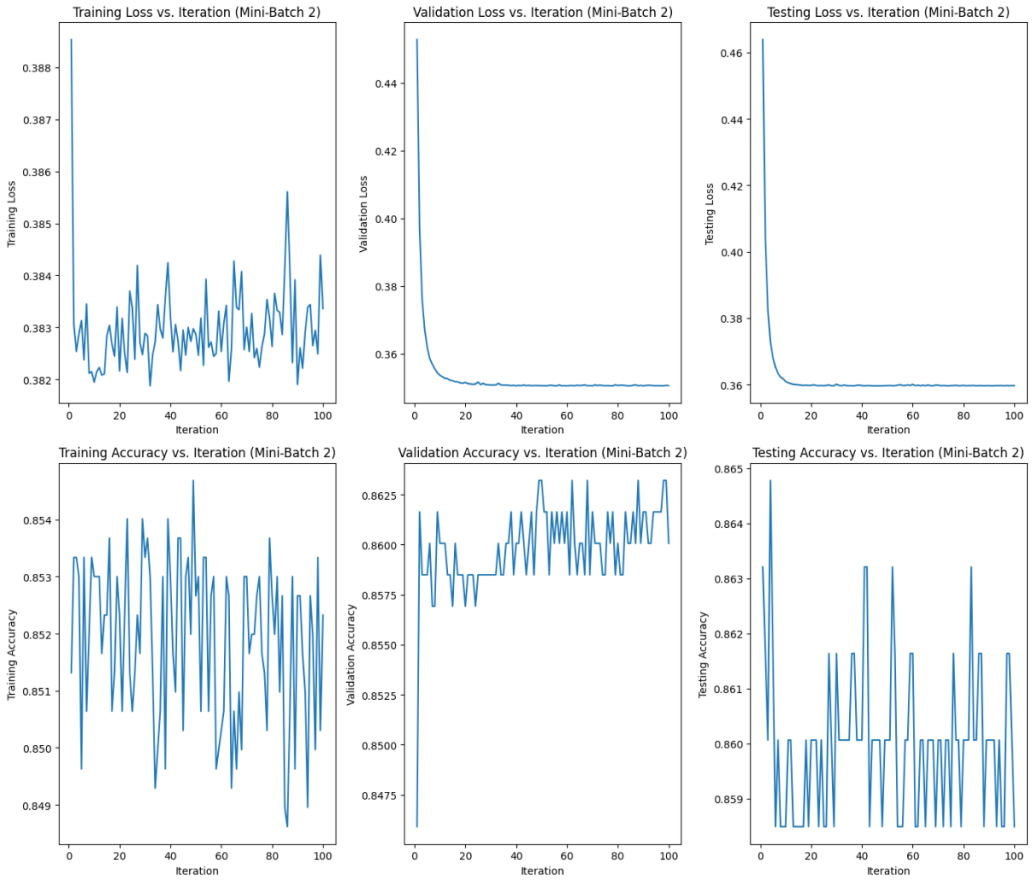
Observations:

- **SGD:** The curves show a general decreasing trend in loss and increasing trend in accuracy, indicating that the model is learning. However, the training and validation curves are quite unstable, suggesting that SGD can be unstable.
- **Convergence speed:** SGD can be relatively slow to converge compared to other optimisation methods, especially for large datasets.

Trade-offs:

- **SGD:**
  - **Pros:** Simple to implement, can be effective for large datasets.

- **Cons:** Can be unstable, slow to converge, sensitive to learning rate.



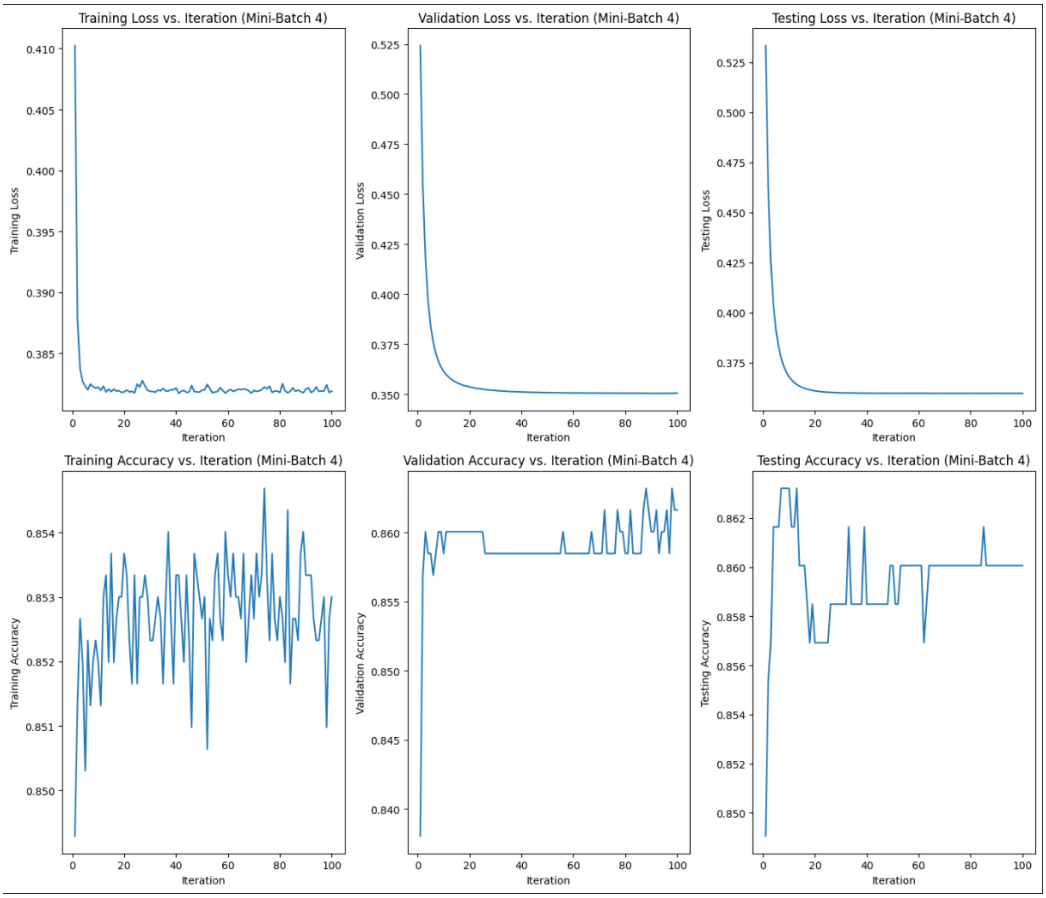
Observations:

- **Mini-batch (Batch size = 2):** The curves show a general decreasing trend in loss and increasing trend in accuracy, indicating that the model is learning. However, the curves are quite volatile, especially in the early iterations.
- **Convergence speed:** Mini-batch gradient descent with a small batch size can be relatively slow to converge compared to larger batch sizes.

- **Stability:** Using a small batch size can introduce more noise into the gradient estimates, making the training process less stable.

Trade-offs:

- **Mini-batch (Batch size = 2):**
  - **Pros:** Can be more efficient than full-batch gradient descent for large datasets.
  - **Cons:** Can be less stable and slower to converge than larger batch sizes.



Observations:

- **Mini-batch (Batch size = 4):** The curves show a general decreasing trend in loss and increasing trend in accuracy, indicating that the model is

learning. The curves are less volatile compared to the smaller batch size(2), suggesting improved stability.

- **Convergence speed:** Mini-batch gradient descent with a batch size of 4 is generally faster to converge than a batch size of 2, as it updates the model parameters more frequently.
- **Stability:** While the stability is improved compared to a smaller batch size, it's still possible for the training process to fluctuate due to the inherent randomness of mini-batch gradient descent.

Trade-offs:

- **Mini-batch (Batch size = 4):**
  - **Pros:** Faster convergence than smaller batch sizes, more stable than smaller batch sizes.
  - **Cons:** Might not be as efficient as larger batch sizes for extremely large datasets.

e. **Model’s performance across different folds:**

Accuracy: Mean = 0.7952755905511811, Std = 0.03591103701099649
Precision: Mean = 0.37908061002178645, Std = 0.04739050464403314
Recall: Mean = 0.5479797979797979, Std = 0.09051982456298951
F1 Score: Mean = 0.4403171933197164, Std = 0.017070600141257355

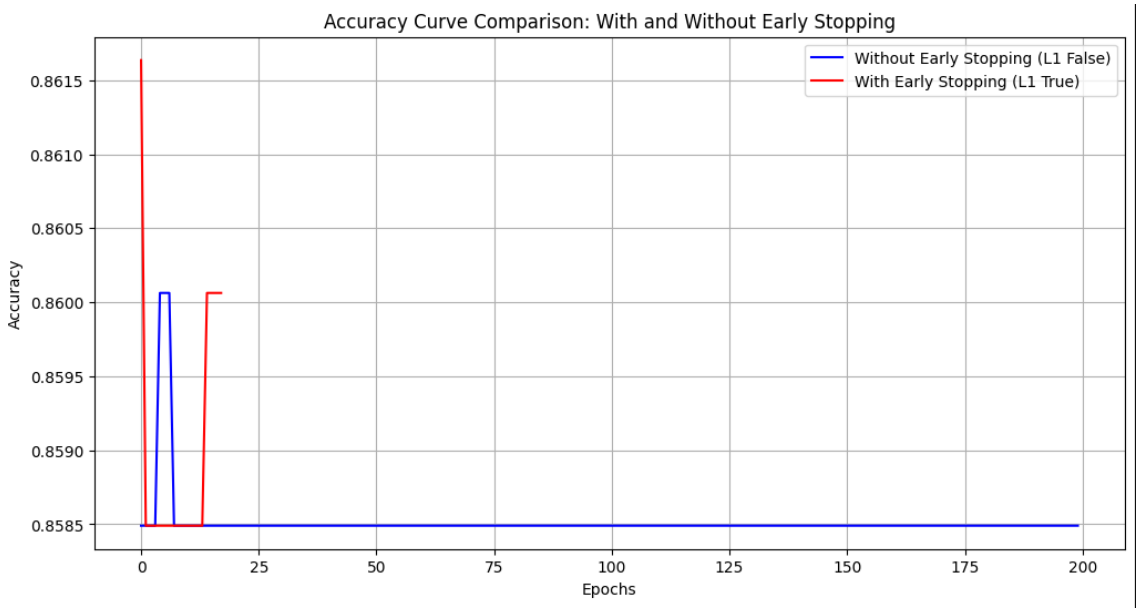
Stability and Variance:

- **Accuracy:** The standard deviation of accuracy is relatively low (0.0359), indicating that the model's accuracy is reasonably stable across different folds.
- **Precision and Recall:** Both precision and recall have higher standard deviations (0.0474 and 0.0905, respectively),

suggesting that the model's performance in terms of identifying positive and negative cases can vary more across folds.

- **F1 Score:** The F1 score has a moderate standard deviation (0.0171), indicating that the model's overall performance is somewhat stable but still exhibits some variation across folds.

f.

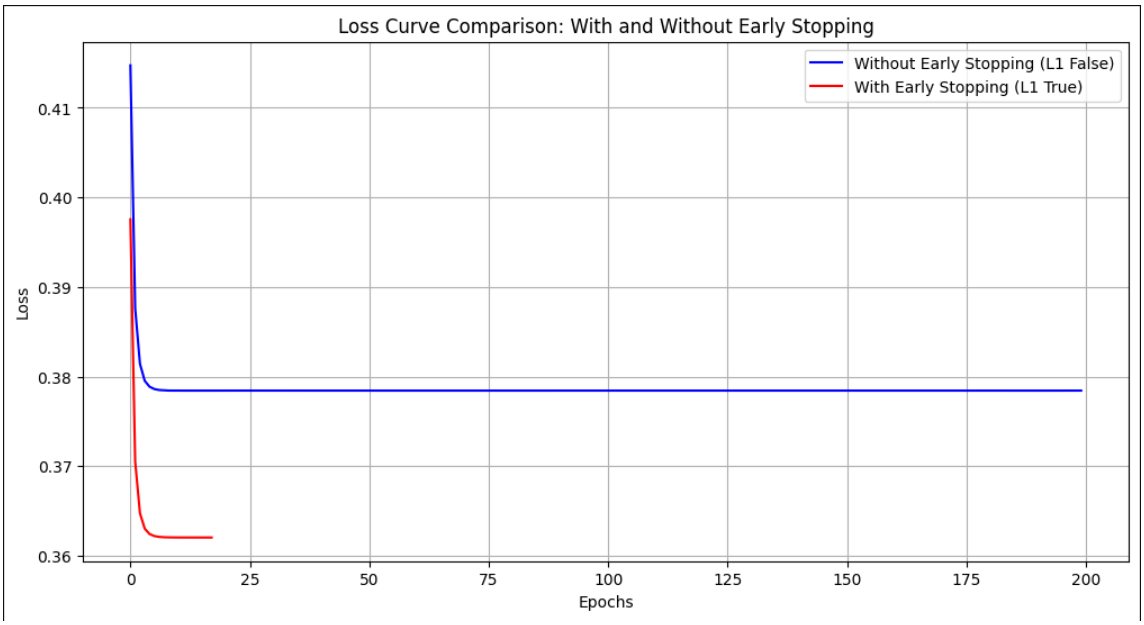


Observations:

- **Without early stopping:** The accuracy curve continues to fluctuate and eventually plateaus at a lower level, suggesting potential overfitting.
- **With early stopping:** The accuracy curve reaches a peak and then starts to decrease, indicating that the model might be starting to overfit. Early stopping prevents further training, leading to a higher final accuracy.

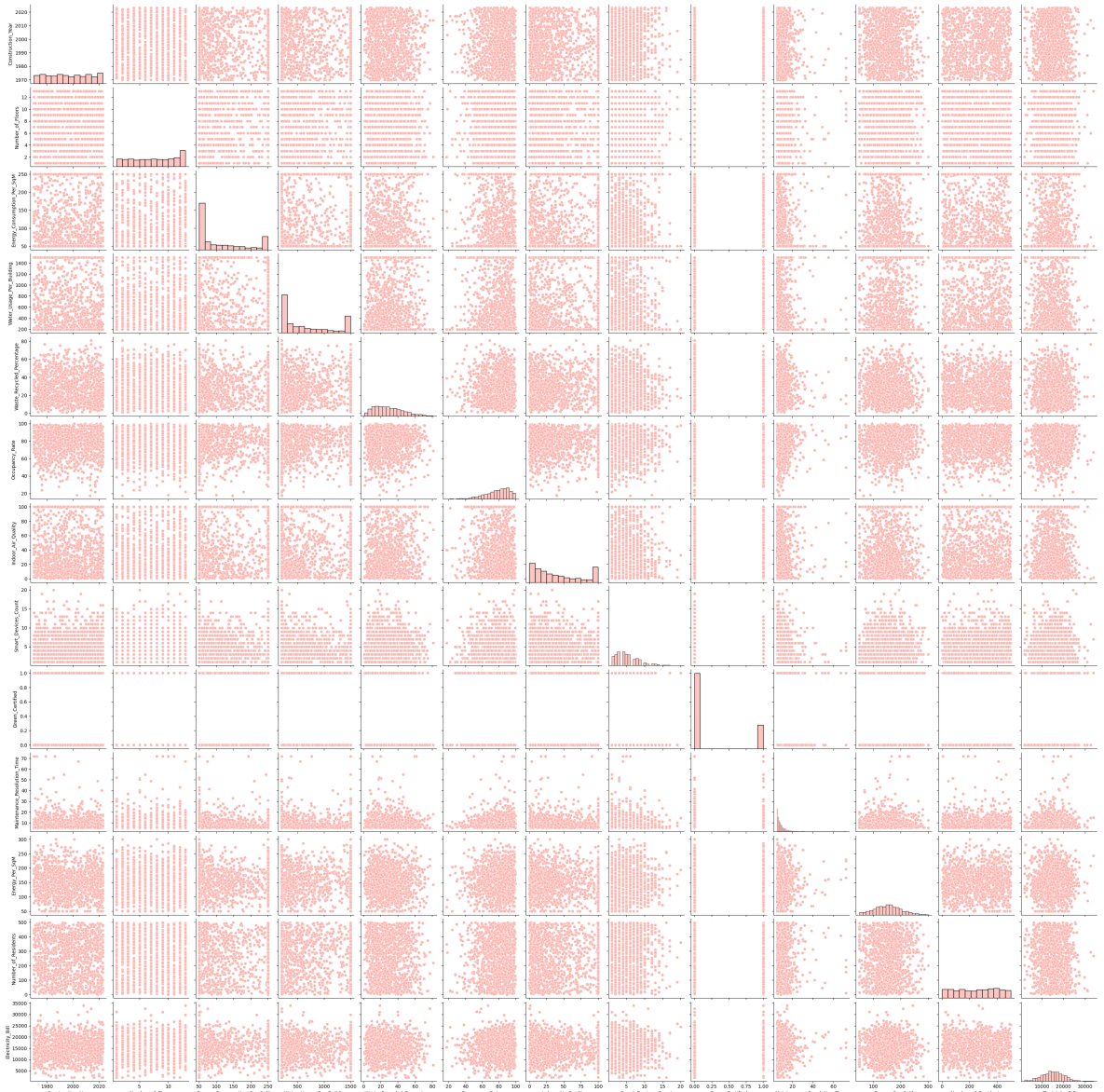
Effect of early stopping:

- **Overfitting:** Early stopping can help prevent overfitting by stopping the training process before the model starts to memorise the training data too much.
- **Generalisation:** By preventing overfitting, early stopping can improve the model's generalisation performance, making it more likely to perform well on unseen data.

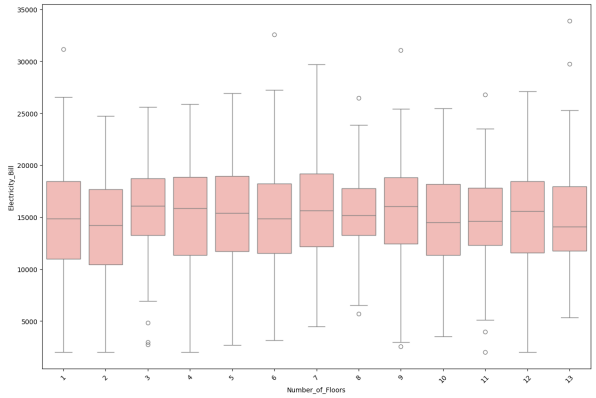


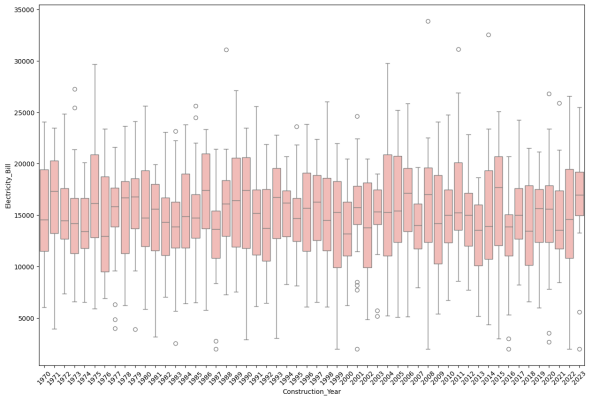
SECTION C

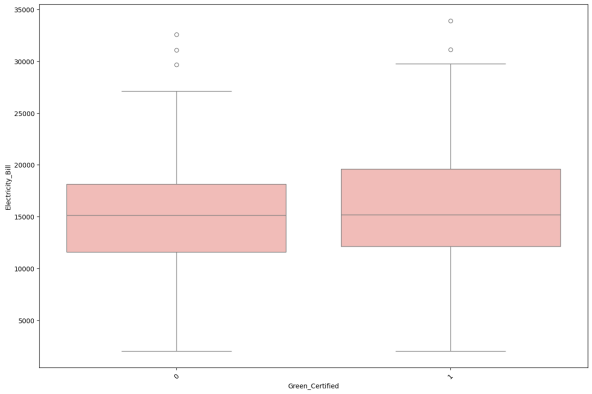
a. **Pair Plot**

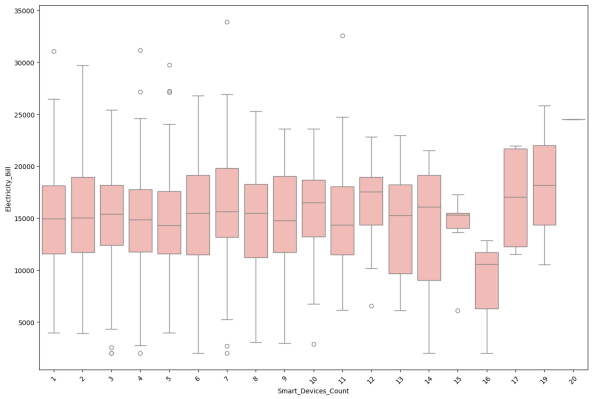

Construction_Year
Number_of_Floors
Energy_Consumption_Per_SqM
Water_Usage_Per_Building
Waste_Recycled_Percentage
Occupancy_Rate
Indoor_Air_Quality
Smart_Devices_Count
Green_Certified
Maintenance_Resolution_Time
Energy_Per_SqM
Number_of_Residents
Electricity_Bill

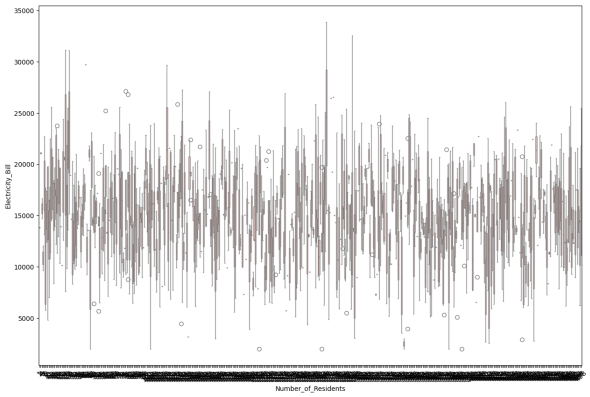
Box Plots


Electricity_Bill
Number_of_Floors

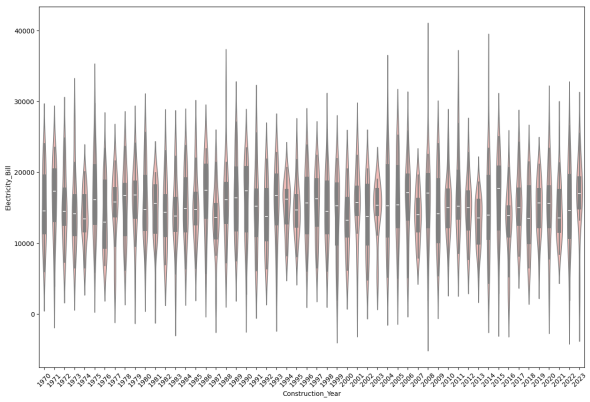


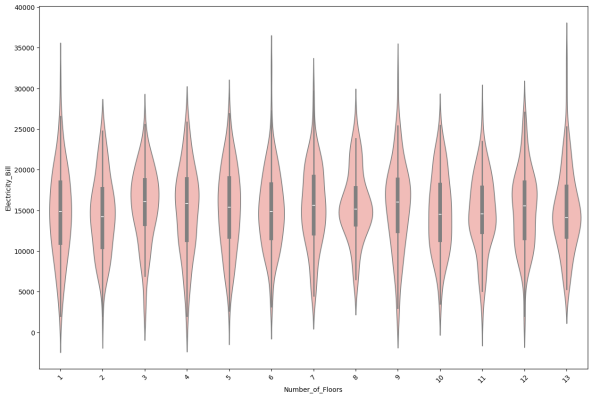


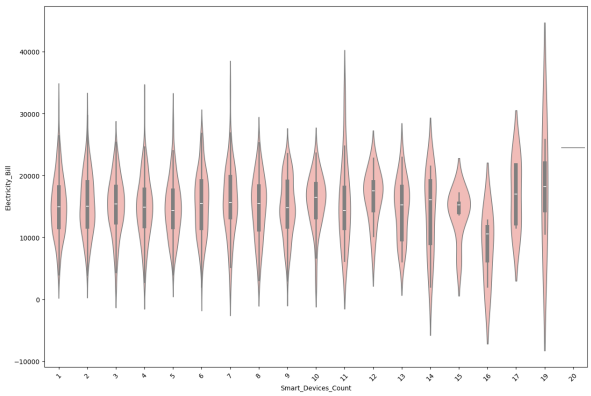


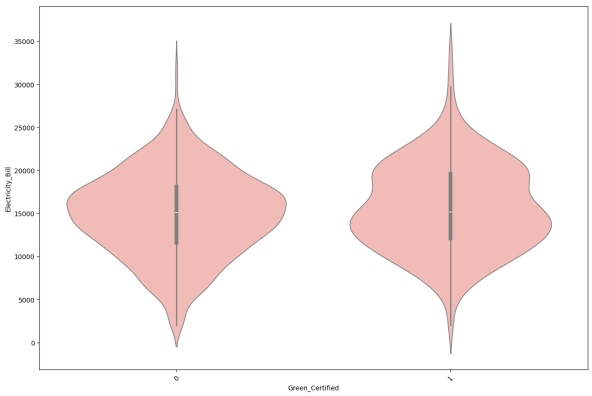


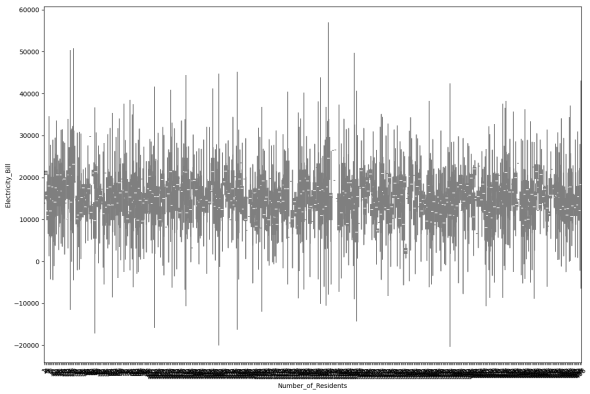
Violin Plots


Electricity_Bill
Construction_Year

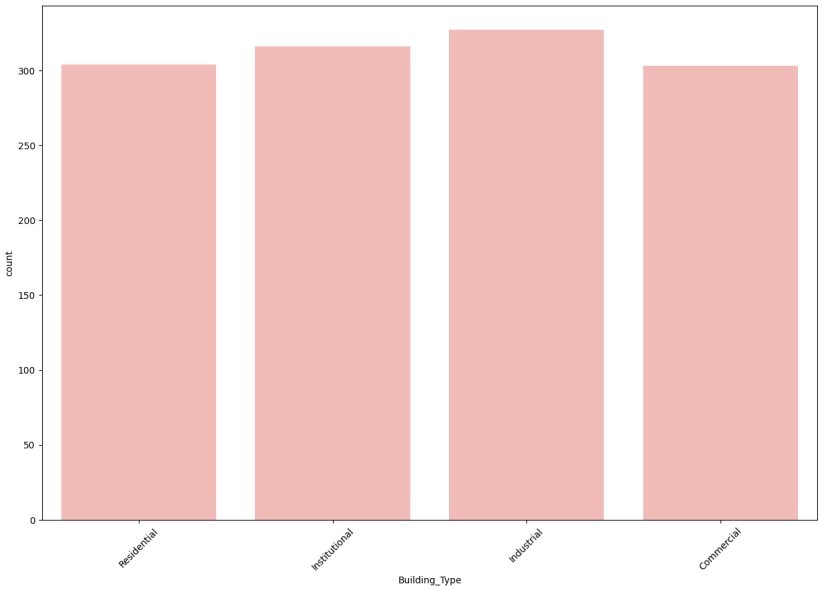

Electricity_Bill
Number_of_Floors

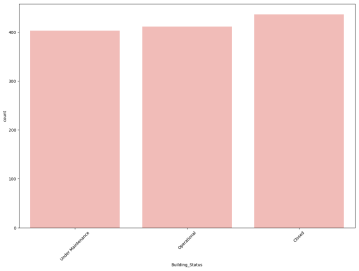

Electricity_Bill
Smart_Devices_Count

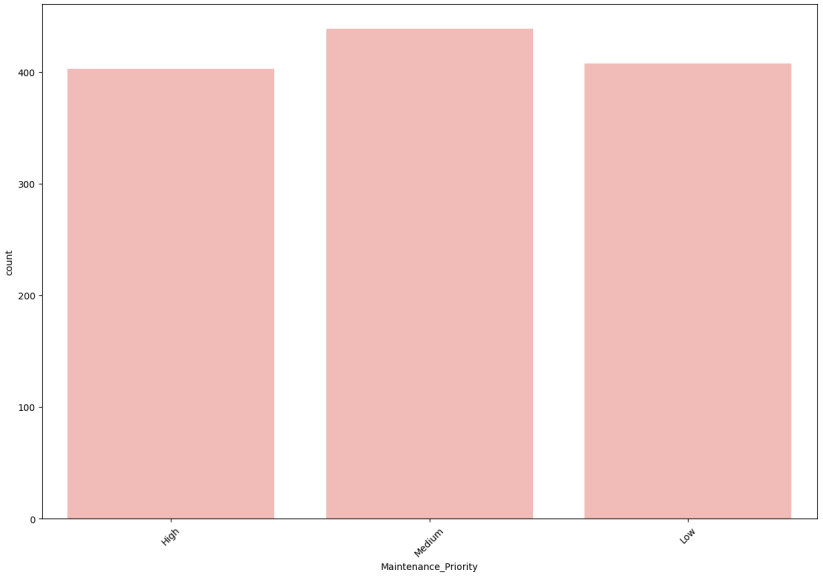

Electricity_Bill
Green_Certified


Electricity_Bill
Number_of_Residents

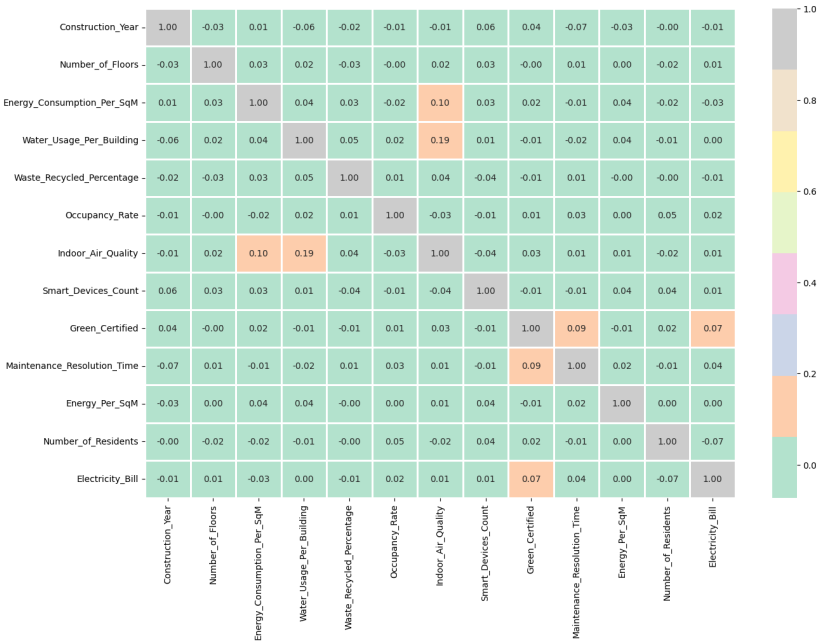
Count Plots







Correlation Heat-map

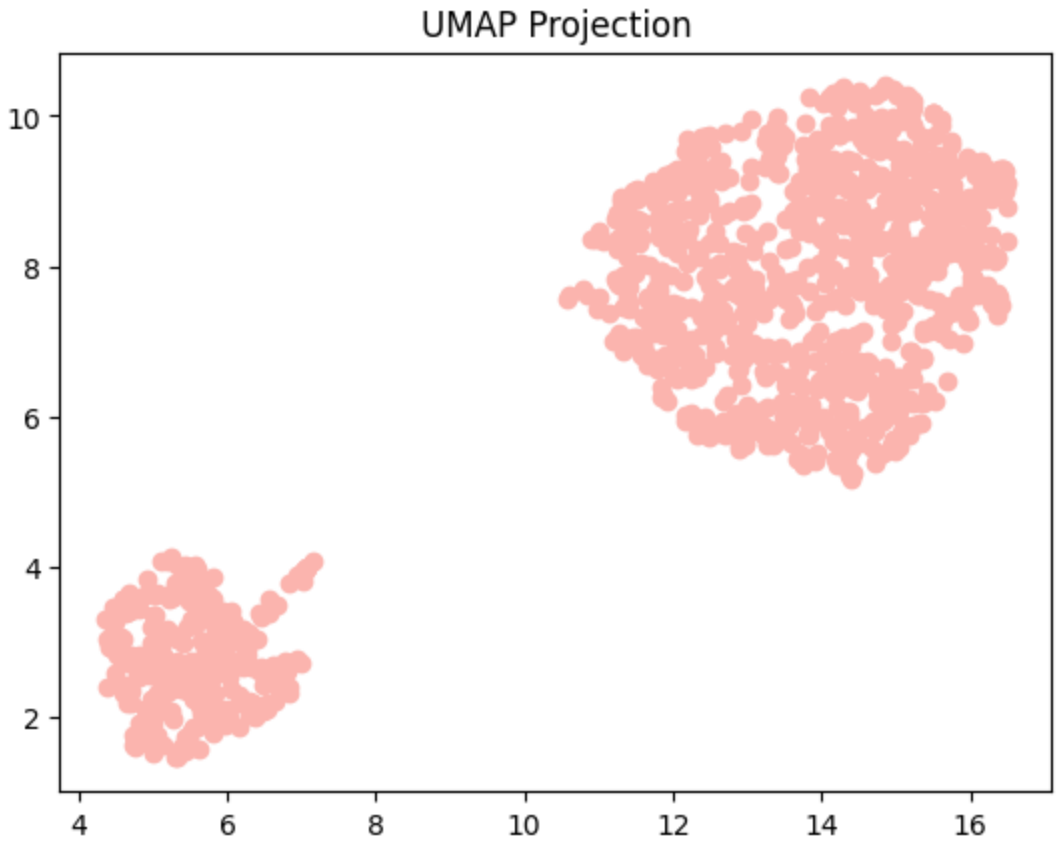


Insights:

- As the number of smart devices count increases, the number of maintenance resolution time decreases.
- The median of electricity bill for all the number of floors is very similar to each other, i.e. around 50,000.
- The median electricity bill is slightly lower for Green Certified buildings compared to non-Green Certified buildings.
- Positive correlations are shown by:
  - Indoor Air Quality and Energy Consumption Per SqM
  - Energy Consumption Per SqM and Electricity Bill
- Negative correlations are shown by:

- Construction Year and Electricity Bill
- Number of Floors and Electricity Bill

b. **Umap Projection**



→ The UMAP scatter plot shows two distinct and well-separated clusters. This indicates that the data has inherent patterns or groupings. The UMAP algorithm has successfully highlighted these underlying structures and relationships in the data.

c. Evaluation metrics for Part c:

Training Set Metrics:

Mean Squared Error (MSE): 24475013.16847547

Root Mean Squared Error (RMSE): 4947.222773281538
R^2 Score: 0.013922520844610209
Adjusted R^2 Score: -0.0011091480449536562
Mean Absolute Error (MAE): 4006.3284693293604

Testing Set Metrics:

Mean Squared Error (MSE): 24278016.155742627
Root Mean Squared Error (RMSE): 4927.272689403604
R^2 Score: 3.734473307526187e-05
Adjusted R^2 Score: -0.06406282547634312
Mean Absolute Error (MAE): 3842.409312558516

d. Evaluation metrics for Part c:

Training Set Metrics:

Mean Squared Error (MSE): 24475013.16847547
Root Mean Squared Error (RMSE): 4947.222773281538
R^2 Score: 0.013922520844610209
Adjusted R^2 Score: -0.0011091480449536562
Mean Absolute Error (MAE): 4006.3284693293604

Testing Set Metrics:

Mean Squared Error (MSE): 24278016.155742627
Root Mean Squared Error (RMSE): 4927.272689403604
R^2 Score: 3.734473307526187e-05
Adjusted R^2 Score: -0.06406282547634312
Mean Absolute Error (MAE): 3842.409312558516

Evaluation metrics for Part d:

Training Set Metrics with Selected Features:

Mean Squared Error (MSE): 24569032.906897984
Root Mean Squared Error (RMSE): 4956.715939702212
R^2 Score: 0.010134545491284008
Adjusted R^2 Score: -0.0011091480449536562
Mean Absolute Error (MAE): 4006.473377514736

Testing Set Metrics with Selected Features:

Mean Squared Error (MSE): 23941409.062998384
Root Mean Squared Error (RMSE): 4892.995918964002
R^2 Score: 0.013901513867940807
Adjusted R^2 Score: -0.06406282547634312
Mean Absolute Error (MAE): 3813.948128176773

→ Selecting the 3 most important features slightly improved the model's performance on the testing set, with decreases in MSE, RMSE, and MAE, while the R^2 score remained almost unchanged. However, the training set metrics worsened slightly. This suggests that feature selection may have led to a better generalisation to unseen data but did not significantly enhance the training performance. The model's overall performance still indicates that it struggles to capture the underlying data patterns effectively.

e. Evaluation metrics for Part c:

Training Set Metrics:

Mean Squared Error (MSE): 24475013.16847547
Root Mean Squared Error (RMSE): 4947.222773281538
R^2 Score: 0.013922520844610209
Adjusted R^2 Score: -0.0011091480449536562
Mean Absolute Error (MAE): 4006.3284693293604

Testing Set Metrics:

Mean Squared Error (MSE): 24278016.155742627
Root Mean Squared Error (RMSE): 4927.272689403604
R^2 Score: 3.734473307526187e-05
Adjusted R^2 Score: -0.06406282547634312
Mean Absolute Error (MAE): 3842.409312558516

Evaluation metrics for Part e:

Training Set Metrics:

Mean Squared Error (MSE): 24188931.13202012
Root Mean Squared Error (RMSE): 4918.224388132379
R^2 Score: 0.025448522951235675
Adjusted R^2 Score: 0.0035036585755213734
Mean Absolute Error (MAE): 3976.7183561620413

Testing Set Metrics:

Mean Squared Error (MSE): 24129096.278434932
Root Mean Squared Error (RMSE): 4912.137648563498
R^2 Score: 0.006171054957969435
Adjusted R^2 Score: -0.09014716879059748
Mean Absolute Error (MAE): 3797.5080199375825

→ **Ridge Regression** slightly outperforms **Linear Regression** in terms of MSE, RMSE, and MAE on both the training and testing sets. Ridge Regression provides marginally better predictive performance and generalises slightly better. However, both models exhibit low R^2 scores, indicating that neither model captures much of the variance in the data. Ridge Regression has a slight advantage in handling potential overfitting due to regularisation, as evidenced by the marginal improvements in metrics.

f. Evaluation metrics for Part f:

Results for 4 components:

MSE Train: 24713262.592420127, MSE Test: 24406551.71736853
RMSE Train: 4971.243565992329, RMSE Test: 4940.298747785252
R^2 Train Score: 0.004323654042929781, R^2 Test Score: -0.005256776527704066
Adjusted R^2 Train: 0.0003209350642079389 4, Adjusted R^2 Test

Score: -0.02166913206285015
MAE Train: 4014.128368168214, MAE Test: 3851.2681146440223

Results for 5 components:

MSE Train: 24667322.37554589, MSE Test: 24495852.12916488
RMSE Train: 4966.620820592799, RMSE Test: 4949.328452342285
R^2 Train Score: 0.006174546336035869, R^2 Test Score: -0.008934880880360874
Adjusted R^2 Train: 0.0011754243357140481, Adjusted R^2 Test Score: -0.02960977598036818
MAE Train: 4013.0389373216444, MAE Test: 3851.044953822016

Results for 6 components:

MSE Train: 24666272.12773407, MSE Test: 24472592.174683336
RMSE Train: 4966.5150888459075, RMSE Test: 4946.978085122607
R^2 Train Score: 0.006216859927764662, R^2 Test Score: -0.007976850137834557
Adjusted R^2 Train: 0.00021212796358205388, Adjusted R^2 Test Score: -0.032865167425188524
MAE Train: 4013.1256526102125, MAE Test: 3848.4130668580265

Results for 8 components:

MSE Train: 24635759.583019286, MSE Test: 24615590.318556298
RMSE Train: 4963.442311845609, RMSE Test: 4961.4101139249005
R^2 Train Score: 0.007446184421600832, R^2 Test Score: -0.013866656072889505
Adjusted R^2 Train: -0.0005663589937645597, Adjusted R^2 Test Score: -0.047521980755806936
MAE Train: 4021.1343355951653, MAE Test: 3868.0686236753063

→ Increasing the number of components in ICA does not significantly improve model performance. While the metrics (MSE, RMSE, MAE) fluctuates slightly, there is no clear improvement in predictive accuracy or generalisation as

indicated by the relatively consistent R^2 scores and Adjusted R^2 scores across different component numbers. The model consistently underperforms with low R^2 and Adjusted R^2 values, suggesting that dimensionality reduction using ICA has limited impact on this dataset's performance.

g. Evaluation metrics for Part g:

Results for alpha = 0.1

MSE Test: 24266701.42894355
RMSE Test: 4926.124382203879
R^2 Test Score: 0.0005033755727057443
Adjusted R^2 Test Score: -0.06356692086494142
MAE Test: 3841.093815452926

Results for alpha = 0.4

MSE Test: 24245407.877904005
RMSE Test: 4923.962619466562
R^2 Test Score: 0.0013804141124682534
Adjusted R^2 Test Score: -0.06263366190596331
MAE Test: 3838.336192362858

Results for alpha = 0.7

MSE Test: 24235588.288183544
RMSE Test: 4922.965395793834
R^2 Test Score: 0.0017848632629884342
Adjusted R^2 Test Score: -0.06220328652784568
MAE Test: 3836.49632693302

Results for alpha = 0.9

MSE Test: 24232651.194060057
RMSE Test: 4922.667081375711
R^2 Test Score: 0.0019058362625798964
Adjusted R^2 Test Score: -0.062074558848793204
MAE Test: 3835.4764941060243

→ Increasing the alpha parameter in ElasticNet regularisation results in a slight improvement in model performance. The metrics show a gradual decrease in MSE, RMSE, and MAE, while R^2 and Adjusted R^2 scores show minimal positive changes. This suggests that a higher alpha value helps improve the model's performance slightly, but the overall impact on the model's generalisation capability remains limited.

h. Evaluation metrics for Part c:

Training Set Metrics:
Mean Squared Error (MSE): 24475013.16847547
Root Mean Squared Error (RMSE): 4947.222773281538
R^2 Score: 0.013922520844610209
Adjusted R^2 Score: -0.0011091480449536562
Mean Absolute Error (MAE): 4006.3284693293604

Testing Set Metrics:
Mean Squared Error (MSE): 24278016.155742627
Root Mean Squared Error (RMSE): 4927.272689403604
R^2 Score: 3.734473307526187e-05
Adjusted R^2 Score: -0.06406282547634312
Mean Absolute Error (MAE): 3842.409312558516

Evaluation metrics for Part g:

Results for alpha = 0.1
MSE Test: 24266701.42894355
RMSE Test: 4926.124382203879
R^2 Test Score: 0.0005033755727057443

Adjusted R^2 Test Score: -0.06356692086494142
MAE Test: 3841.093815452926

Results for alpha = 0.4

MSE Test: 24245407.877904005
RMSE Test: 4923.962619466562
R^2 Test Score: 0.0013804141124682534
Adjusted R^2 Test Score: -0.06263366190596331
MAE Test: 3838.336192362858

Results for alpha = 0.7

MSE Test: 24235588.288183544
RMSE Test: 4922.965395793834
R^2 Test Score: 0.0017848632629884342
Adjusted R^2 Test Score: -0.06220328652784568
MAE Test: 3836.49632693302

Results for alpha = 0.9

MSE Test: 24232651.194060057
RMSE Test: 4922.667081375711
R^2 Test Score: 0.0019058362625798964
Adjusted R^2 Test Score: -0.062074558848793204
MAE Test: 3835.4764941060243

Evaluation metrics for Part h:

Training Set Metrics:

Mean Squared Error (MSE): 14926446.25730777
Root Mean Squared Error (RMSE): 3863.4759294329465

R^2 Score: 0.398626166333897
Adjusted R^2 Score: 0.38945888228410885
Mean Absolute Error (MAE): 3092.7481886865007

Testing Set Metrics:

Mean Squared Error (MSE): 24470497.24090483
Root Mean Squared Error (RMSE): 4946.766341854528
R^2 Score: -0.007890564029022329
Adjusted R^2 Score: -0.07249893351806214
Mean Absolute Error (MAE): 3812.046396052472

→ The Gradient Boosting Regressor shows significant improvement on the training set compared to linear regression and ElasticNet models, with a lower MSE and higher R^2 score. However, its performance on the testing set is slightly worse than the ElasticNet models in terms of MSE, RMSE, and MAE, and it still shows a negative R^2 score, indicating poor generalisation. This suggests that while Gradient Boosting Regressor may fit the training data better, it does not significantly outperform ElasticNet in generalising to new, unseen data.