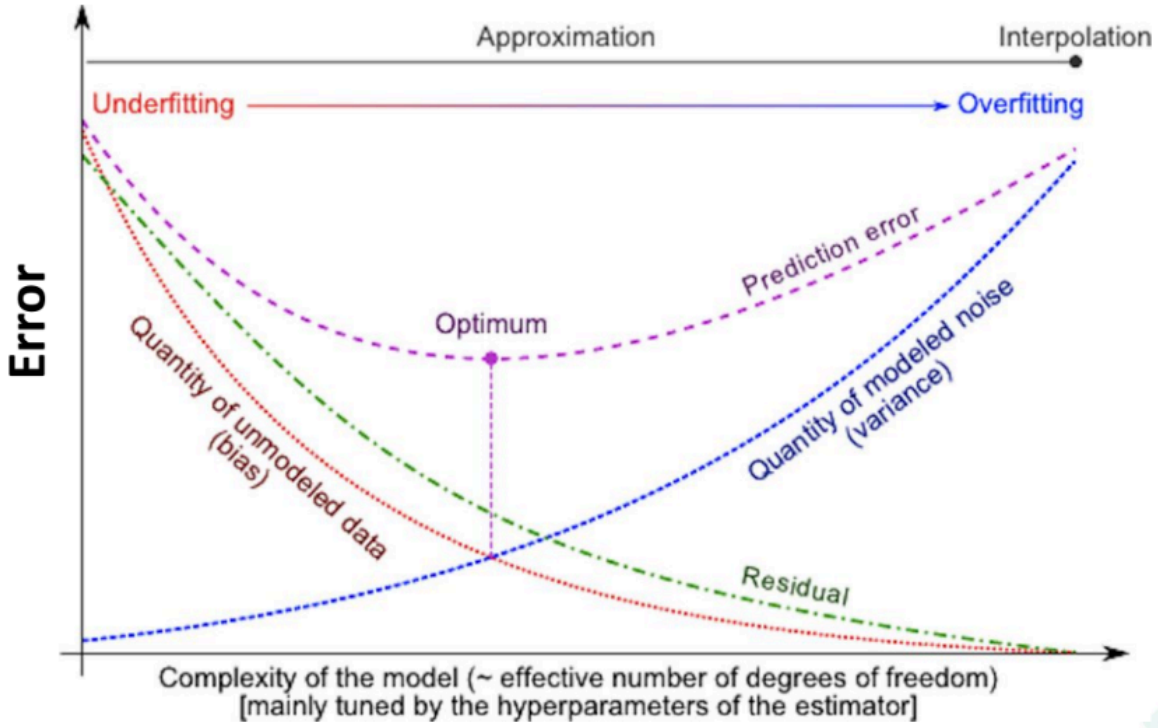
Question 1

a) When we are developing a machine-learning model for prediction tasks and increasing the complexity of the model by adding more features or including higher-order polynomials, then this has a significant effect on its performance especially in terms of bias and variance. In the below provided graph we can divide it into 3 sections, when the complexity is low, when it is Optimal and when it is high.



Low complexity - High Bias and Low Variance. Which means that the model is too simple to make good approximations about the data i.e predictions are consistent but inaccurate(underfitting).

Optimal Complexity - Balanced Bias and Variance. Here the performance of our model will be optimum, it will provide both accurate and consistent predictions.

High Complexity - Low Bias and High Variance. Here the model will be overfitting the dataset i.e its predictions will be accurate but only on seen data. Thus it will be inconsistent on unseen or new data.

b) To measure the performance of the model we will use several metrics : accuracy, precision, recall and F1 score. Here the given data is :-

1. True Positives (TP) :- 200 (correctly identified spam)
2. False Negatives (FN) :- 50 (spam classified as legitimate)

3. True Negatives (TN) :- 730 (correctly identified as legitimate)
4. False Positives (FP) :- 20 (legitimate classified as spam)

Accuracy = (TP + TN) / (TP +TN + FP + FN) = (200+730) / (200 + 730 + 20 + 50) = 0.93
Precision = (TP) / (TP + FP) = (200) / (200 + 20) = 0.909
Recall = (TP) / (TP + FN) = (200) / (200 + 50) = 0.80
F1 Score = 2 * [(Precision * Recall) / (Precision + Recall)] = 2 * [(0.80 * 0.909) / (0.80 + 0.909)] = 2 * [(0.73) / (1.71)] = 2 * [0.4267] = 0.851

d). Model f1 - lower empirical risk on the training set than model f2, but may not generalize better. To understand it we'll use the concept of overfitting and underfitting.

TOY EXAMPLE SETUP

Training data - Features(X) = {1,2,3} and Labels(Y) = {2,4,6}
This is a simple linear relationship Y = 2X, and we'll be using two models to predict the labels.

Model F1(complex model, overfitting) :- Since model F1 is a high-degree polynomial, this has the capacity to overfit the training data. In this case, F1 will interpolate all the points in the training data leading to zero training error. Since it's overfitting, thus empirical risk(training loss) is zero.

Model F2(Simple model, Underfitting) :- Let F2 be a simple linear model, such that F2(X) = 2X. This model does not overfit the data rather captures the true linear relationship between X and Y. This model will have small training errors due to some noise in the dataset or some slight variations. However, the empirical risk will be non-zero but small.

Thus we can say that, the Model F1 will have lower empirical risk on the training set but may perform poorly on new data i.e. will have poor generalization due to overfitting.
Model F2 will have a slightly higher empirical risk on the training set but may generalize better because it captures true pattern in the data without overfitting.

DATE:
PAGE:

Q c) The equation of a simple linear Regression is given by :-

$$y = mx + c$$

m → slope, c → intercept

to calculate slope & intercept

$$(m) \Rightarrow \frac{n\Sigma xy - \Sigma x \Sigma y}{n\Sigma x^2 - (\Sigma x)^2}$$

$$(c) \Rightarrow \frac{\Sigma y - m\Sigma x}{n}$$

$$\therefore m \Rightarrow \frac{5(45+180+550+1275+1800) - (52 \times 285)}{5 \times 694 - (52)^2}$$

$$m \Rightarrow \frac{5(3850) - 14820}{3870 - 2704} \Rightarrow \frac{4430}{766}$$

$$\boxed{m \Rightarrow 5.78}$$

$$c \Rightarrow \frac{285 - (5.78)(52)}{5} \Rightarrow \frac{285 - 300.56}{5}$$

$$c \Rightarrow \frac{-15.56}{5} \Rightarrow \boxed{-3.11}$$

$\therefore y = mx + c$

at $x = 12$

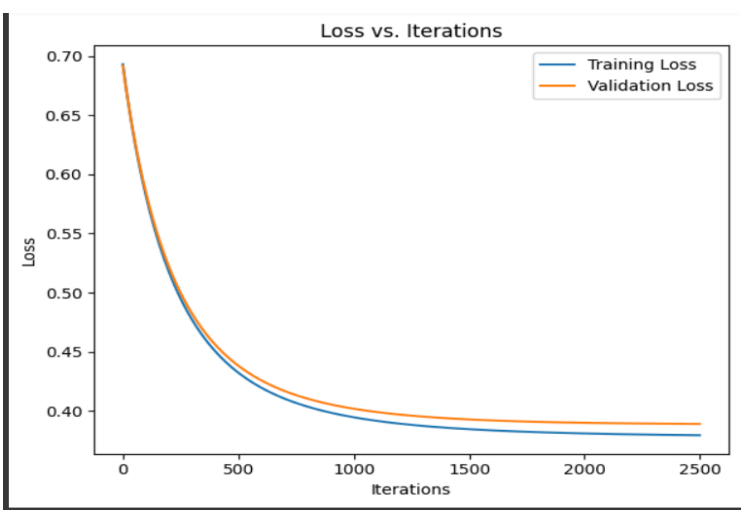
$y = 5.78 \times 12 + (-3.11)$

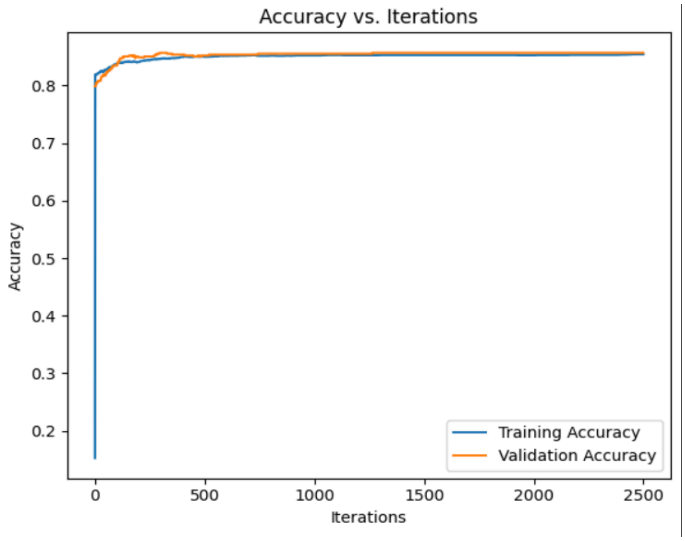
$y \Rightarrow 69.36 - 3.11$

$$\boxed{y = 66.25}$$

SECTION B

PART A).





Analysis of the results

Convergence of the Model :- The loss plots show that the model is converging. We can see that Both training and Validation losses decrease after a few hundred iterations and then start to stabilize. This is an indication that the model is learning from the data.

Comparison of training and validation metrics :-

- **Loss :** The training loss decreases more rapidly and reaches a lower value as compared to the validation loss. This is expected as the model is optimizing for the training data.
- **Accuracy :** Both the training and validation accuracy increases over time, which is a good sign. The training accuracy is slightly higher than the validation accuracy which is expected.

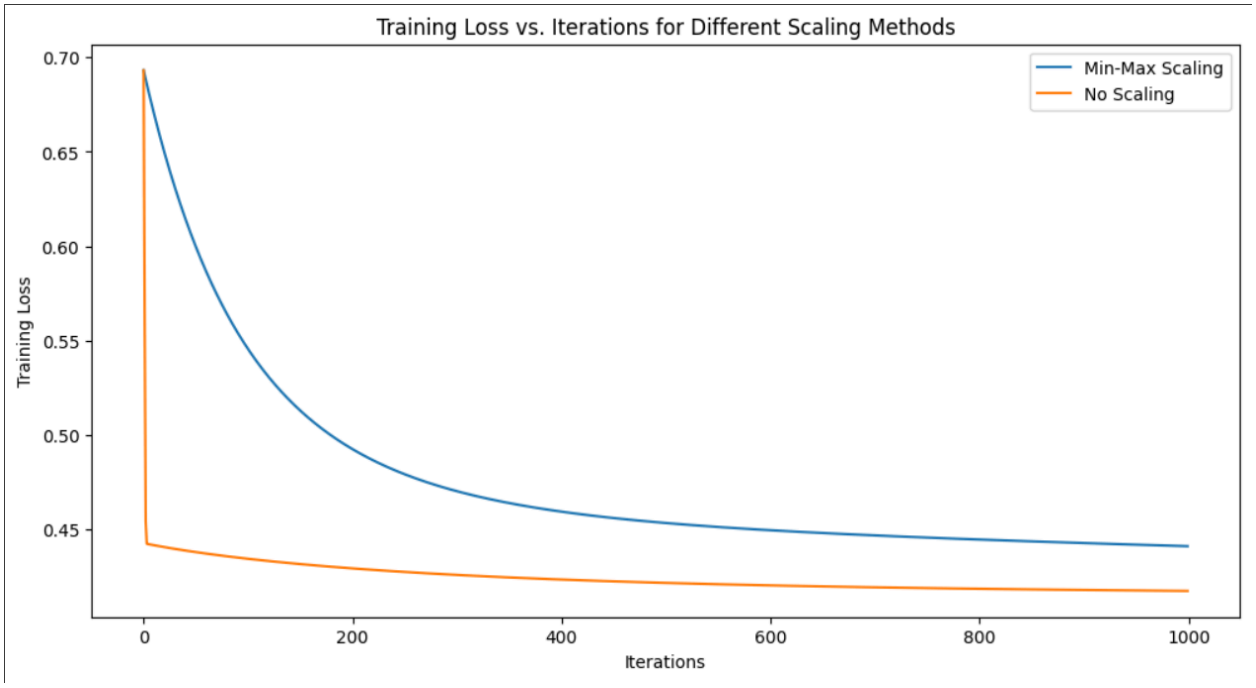
PART B).
Observations

- **Min-Max Scaling (Blue Curve) :** we can see that the loss decreases steadily, and convergence is achieved over time.
- **No Scaling (Orange Curve) :** Here we can see that the loss decreases sharply early in the training of the model, but it flattens sooner than the scaled version.

General Observation on Feature Scaling :

- In gradient-based algorithms like logistic regression, feature scaling is critical for improving the convergence speed and stability. It helps prevent large differences in feature ranges from causing inefficient or biased gradient updates.
- Without scaling, the model can converge prematurely to a suboptimal solution, as we can also see here in the case of no scaling, the loss converges faster, which is an indication of poor generalization.

Thus we can say that Min-Max scaling leads to a more stable and consistent convergence, ensuring the model reaches a better solution after sufficient iterations. Also we can conclude by saying that scaling features tends to improve the overall model convergence and its performance, making it an essential step during the preprocessing of the data.



PART c)

Precision: 0.8 **Recall:** 0.08247422680412371

F1 Score: 0.14953271028037382 **ROC-AUC Score:** 0.5393818258324885

Comment on how these metrics provide insight into the model's performance.

- **Precision (0.8) :** This indicates that 80% of the positive predictions made by the model are correct. High precision means whenever my model predicts a positive class, usually it is correct.
- **Recall (0.082) :** The recall score I've got is quite low, meaning that the model only correctly identifies about 8.2% of the actual positive cases.

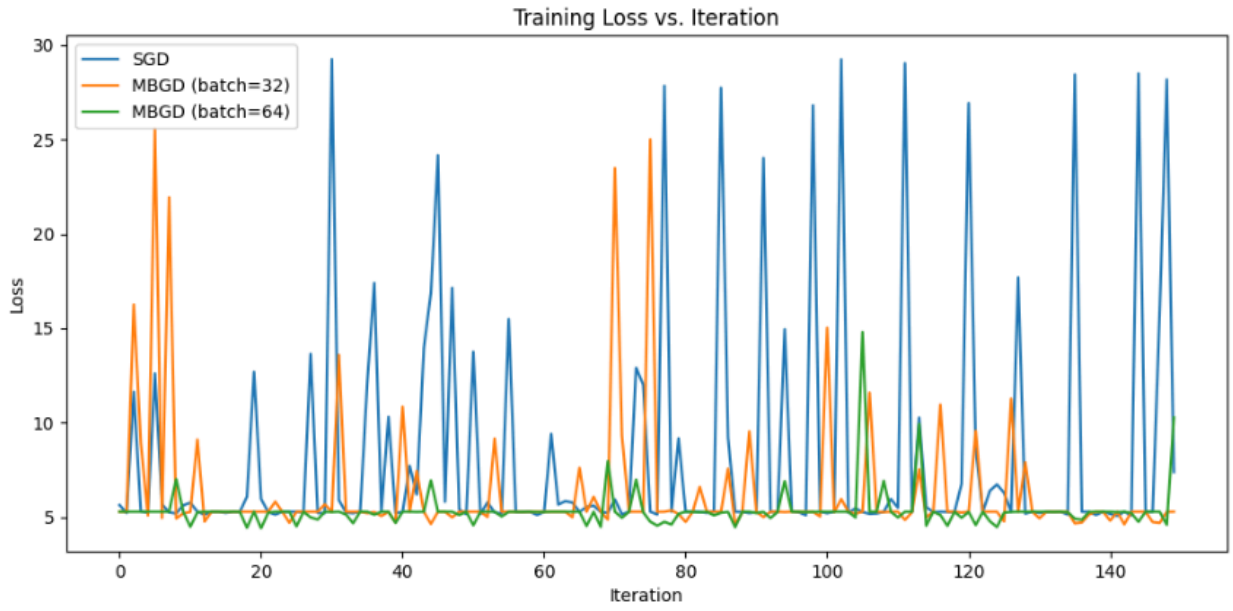
- **F1 Score (0.150) :** The F1 Score is the harmonic mean of precision and recall. Since the recall is low, the F1 score is also low, indicating a balance between precision and recall is not optimal. This means that model fails to predict positive cases while maintaining precision.
- **ROC-AUC Score(0.54) :** The roc score measures the model's ability to distinguish between classes. A score of 0.5 indicates random guessing, thus a score of 0.54 is only slightly better than random. This means that the model's overall ability to differentiate between positive and negative classes is weak.

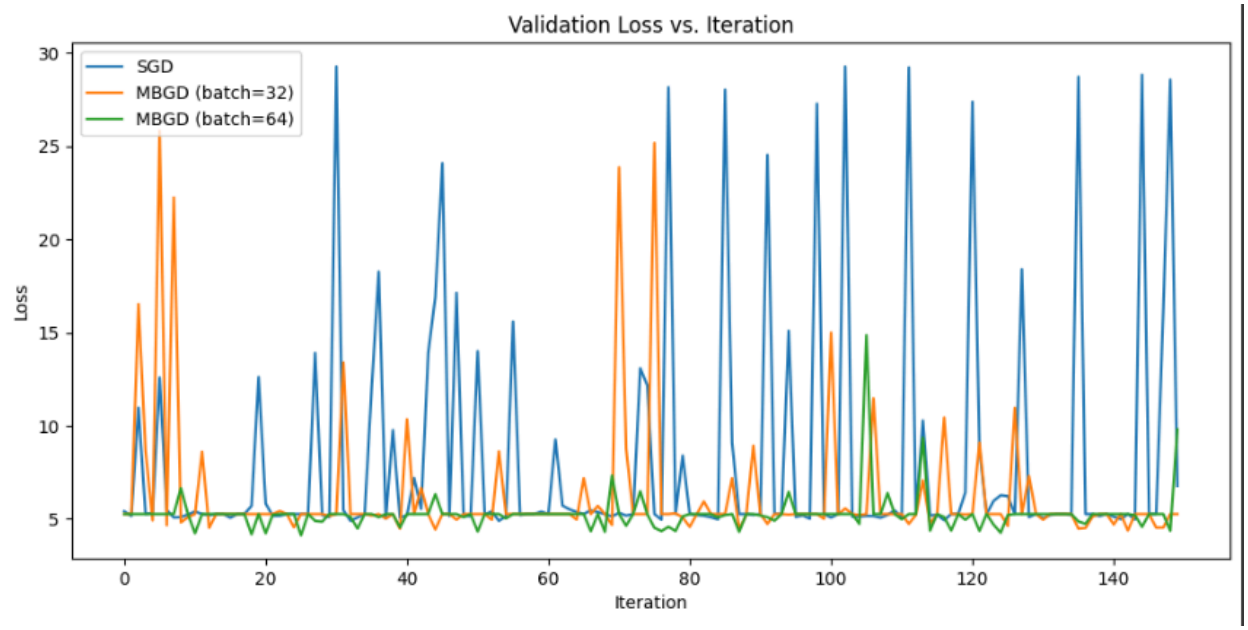
PART D).

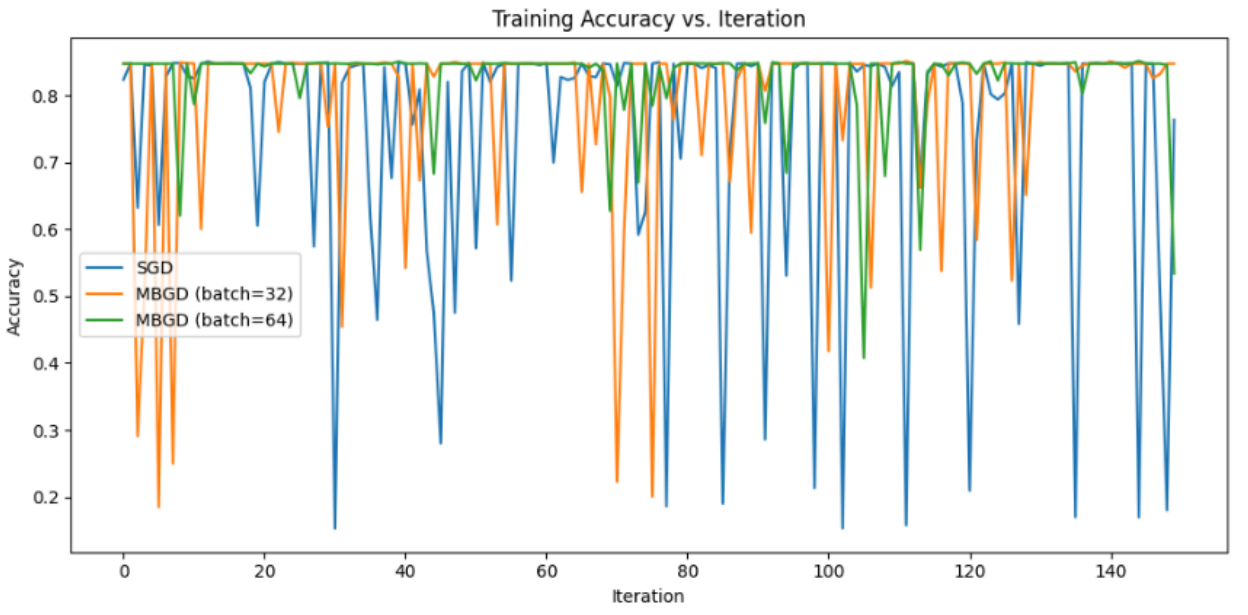
<u>Discuss the trade-offs in terms of convergence speed and stability between these methods.</u>

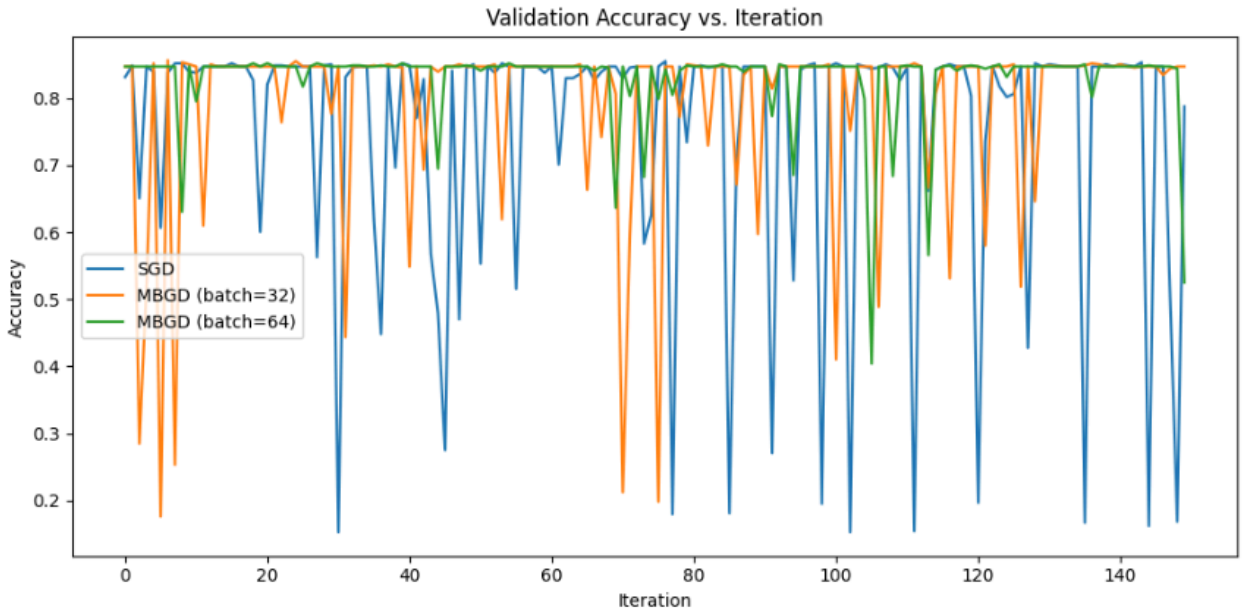
- **Convergence Speed :** Here in the plots we can see that the SGD shows the fastest initial progress, as it updates parameters more frequently. Whereas MBGD converges slower than SGD.
- **Stability :** SGD's updates are noisy, which can help escape local minima but may never 'settle' at the global minimum. Whereas MBGD offers a balance, reducing noise compared to SGD.

In the plots we can see that the SGD shows more oscillations in both loss and accuracy curves, whereas MBGD with larger batch size (64) have a more smoother curve as compared to SGD and MBGD with small batch size (32).




Validation Loss vs. Iteration
SGD
MBGD (batch=32)
MBGD (batch=64)
Loss
Iteration
30
25
20
15
10
5
0
20
40
60
80
100
120
140


Training Accuracy vs. Iteration
SGD
MBGD (batch=32)
MBGD (batch=64)
Accuracy
Iteration
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80
100
120
140



PART E).

Average Accuracy: 0.8285, **Std Dev:** 0.0421

Average Precision: 0.1894, **Std Dev:** 0.2775

Average Recall: 0.0880, **Std Dev:** 0.1258

Average F1 Score: 0.0943, **Std Dev:** 0.1176

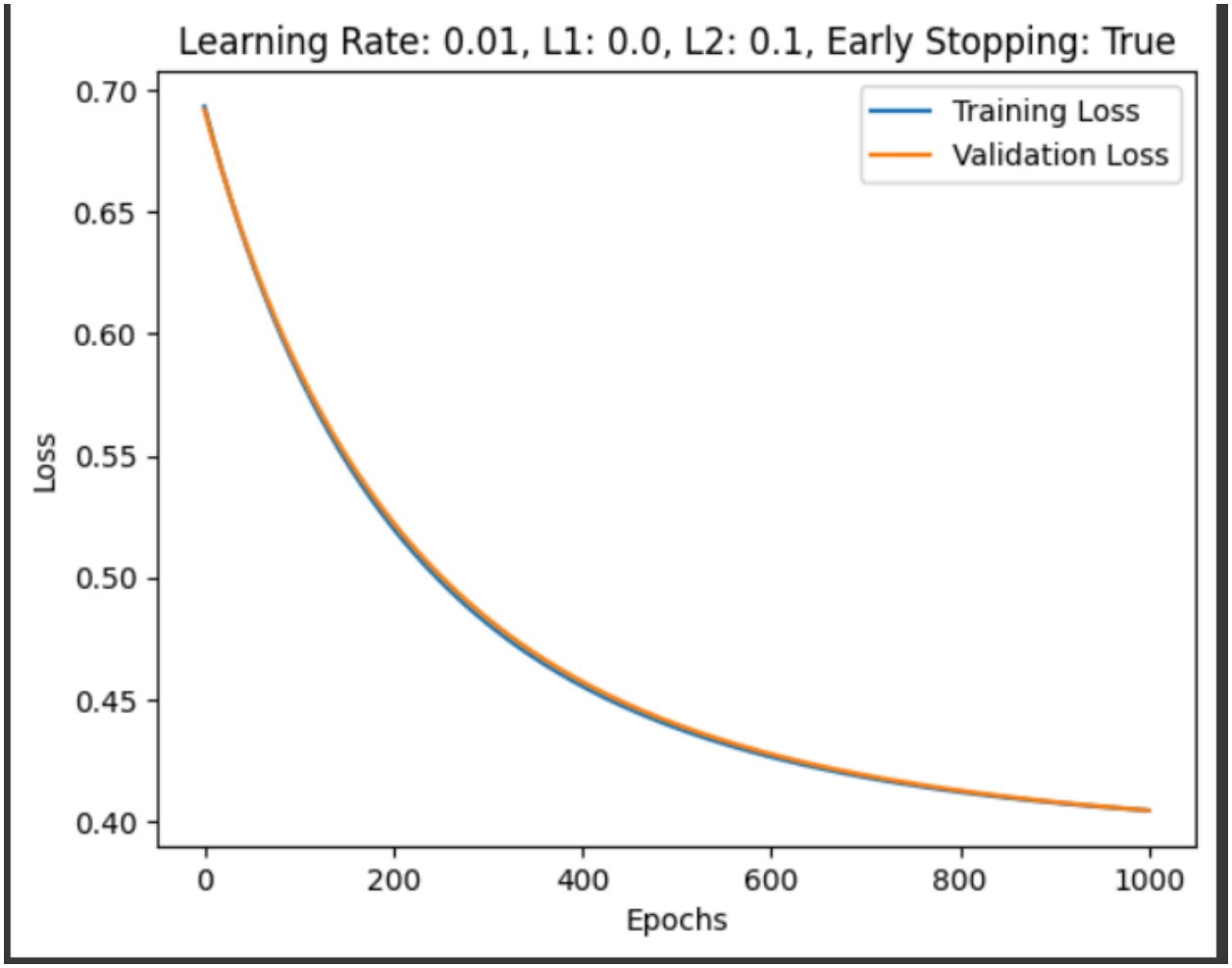
Discussion on the stability and variance of the model's performance across different folds.

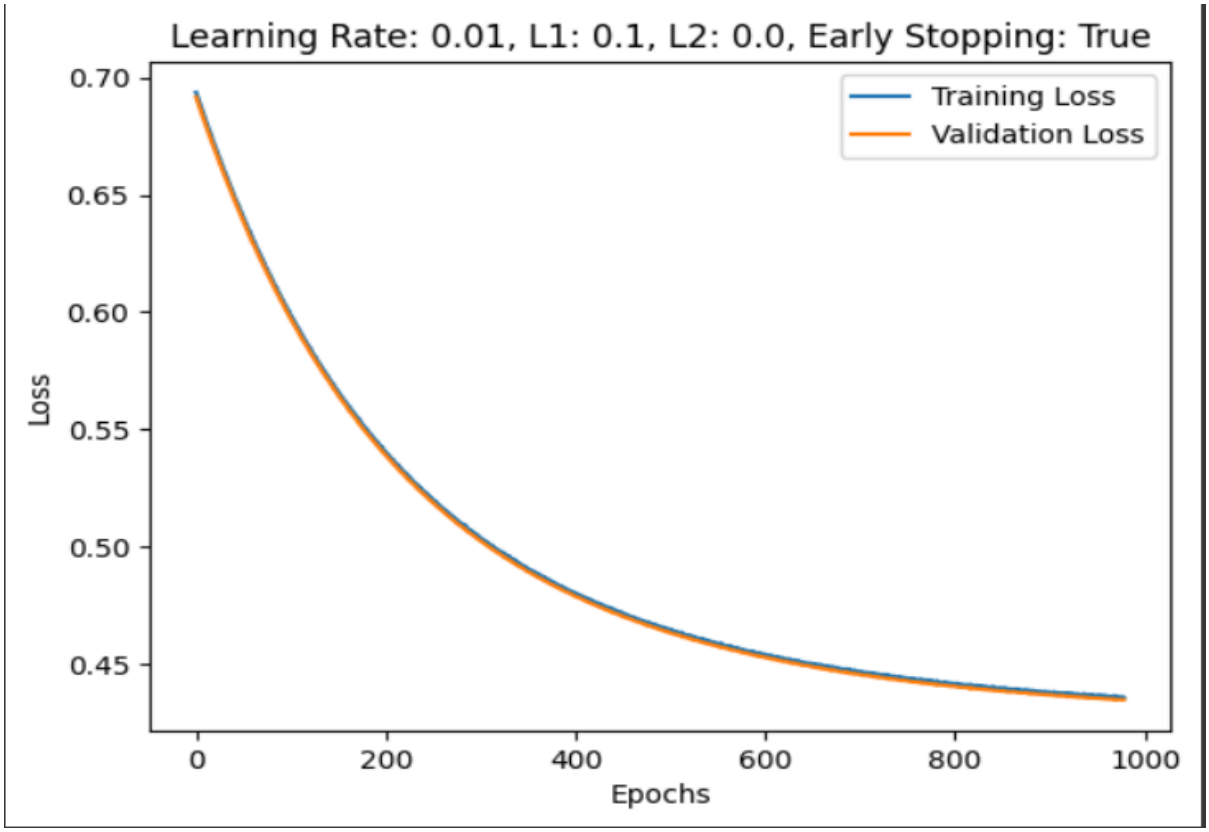
These metrics provide the insights into the stability and variance of the model's performance across different folds in cross-validation:

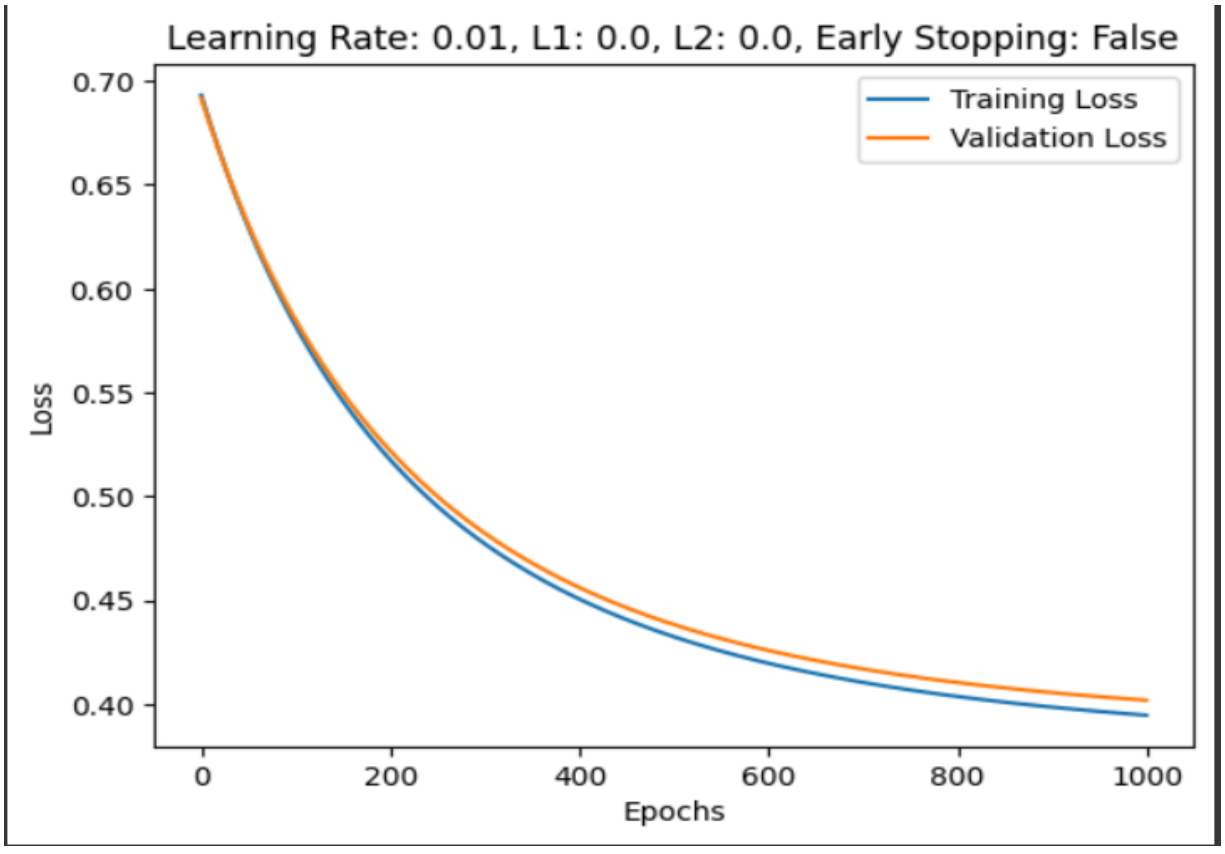
- **Average Accuracy (0.83, Std Dev - 0.042) :** The average accuracy is relatively high, which suggests that the model performs well overall. The Std dev is also low, suggesting that the model's accuracy is consistent across different folds.
- **Average Precision(0.19, Std Dev - 0.28) :** The avg precision score is quite low, and the high std deviation indicates significant variation in precision across different folds. This tells that the model's ability to correctly identify positive instances varies considerably depending on the fold.
- **Average Recall (0.089, Std Dev - 0.123) :** The average recall is very low, indicating that the model misses many positive instances. The std deviation is also high suggesting that recall varies significantly across folds.

- **Average F1 Score (0.094, Std Dev - 0.117) :** The F1 score, which balances precision and recall, is low on average. The high std deviation shows that the f1 score the balance between precision and recall across different folds is inconsistent.

PART F).




Learning Rate: 0.01, L1: 0.1, L2: 0.0, Early Stopping: True
Training Loss
Validation Loss
0.70
0.65
0.60
0.55
0.50
0.45
Loss
0
200
400
600
800
1000
Epochs


Learning Rate: 0.01, L1: 0.0, L2: 0.0, Early Stopping: False
Training Loss
Validation Loss
0.70
0.65
0.60
0.55
0.50
0.45
0.40
Loss
0
200
400
600
800
1000
Epochs

First and second plot with early stopping

- **Observation :** The training and validation losses are very closely aligned throughout the training process. Early stopping occurs at the 979th iteration when it detects that validation loss is no longer improving,preventing the model from overfitting.
- **Effect on overfitting :** Since training stops early when validation loss stagnates, thus the model is less likely to overfit.
- **Effect on Generalization :** improved generalization, as early stopping ensures that model is well-fitted to the training data without overtraining, leading to performance better on unseen data.

Third plot- without early stopping

- **Observation :** The training loss continues to decrease as the number of epochs increases, while validation loss starts to deviate slightly from the training loss in later epochs. The validation loss is higher than the training loss, suggesting the model is beginning to overfit, especially at later stages of training.
- **Effect on Overfitting :** Overfitting is more likely in this case because the model continues to train beyond the point where it performs best on the validation set. The divergence in validation and training loss suggests that the model is fitting more on training data at expense of performance on unseen data.
- **Effect on Generalization :** Weaker generalization, as the model is prone to overfitting, which leads to poor performance on unseen data.

SECTION - C

PART A).

five detailed insights based on the EDA visualizations:

- **Energy Consumption Patterns:** The pair plots likely reveal relationships between Energy Consumption Per SqM and other features like Building Type and Construction Year. For instance, newer buildings might show lower energy consumption due to better insulation and energy-efficient designs.
- **Outliers in Data:** The box plots help identify outliers in numeric columns. For example, Electricity Bill and Energy Consumption Per SqM might have significant outliers, indicating buildings with unusually high energy usage or billing errors.
- **Distribution of Features:** The violin plots provide insights into the distribution of numeric features. Features like Water Usage Per Building and Waste Recycled Percentage might show skewed distributions, indicating that most buildings fall within a certain range, with a few outliers.
- **Categorical Feature Trends:** The count plots for categorical features such as Building Type and Maintenance Priority highlight the frequency of each category. This can reveal trends like the most common building types or the priority levels most frequently assigned to maintenance tasks.
- **Correlation Insights:** The correlation heatmap shows the relationships between numeric features. High correlations between Energy Consumption Per SqM and Electricity Bill suggest that energy consumption is a significant factor in determining electricity costs. Additionally, features like Number of Residents and Occupancy Rate might also show strong correlations with energy and water usage.

You can see all the plots in the ipynb notebook, since there were many plots, thus i didn;t include them in my report.

PART B).

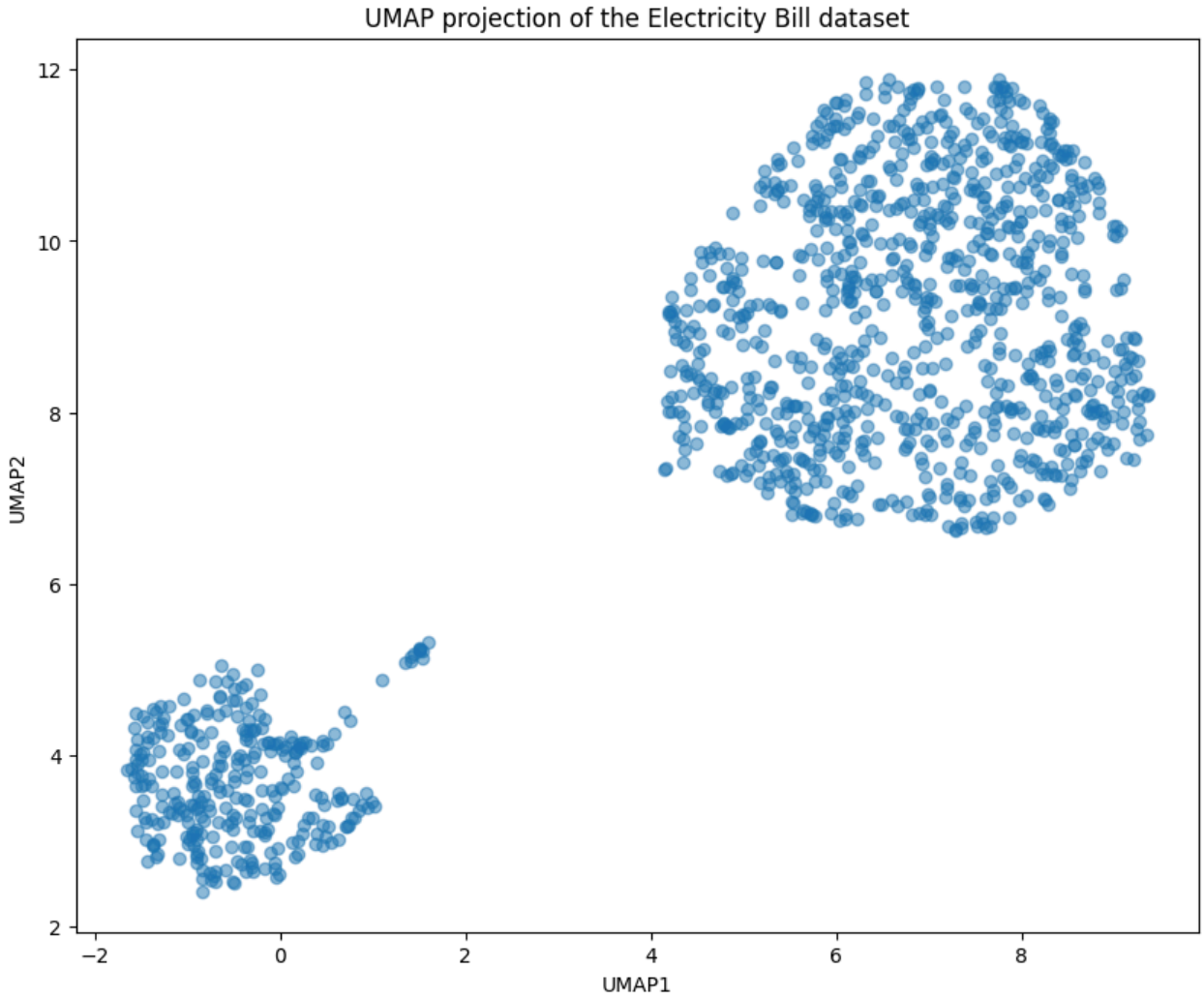
Comment on the separability and clustering of the data after dimensionality reduction.

Below you can see the UMAP scatter plot of the Electricity Bill dataset, few observations on the separability and clustering are as follows:

- **Distinct Clustering :** The plot clearly shows two distinct clusters of data points. One cluster is dense and occupies the central top-right portion of the plot, while the other cluster is smaller and located towards the bottom left corner. This indicates good separability between the two groups.
- **Clear Gap :** There is a visible gap between the two clusters, suggesting that the UMAP algorithm has effectively separated the data into distinct groups.

- **Cluster Density :** The larger cluster is more densely packed, suggesting that the data points within this cluster are very similar to each other. The smaller cluster is less dense, indicating more variability among the data points in this group.
- **Outliers :** There do not appear to be any significant outliers in the plot, as most data points are grouped within the two main clusters. This suggests that the dataset is relatively clean and well-structured.

Overall, the UMAP projection has successfully reduced the data dimensions while preserving the inherent structure, resulting in well-separated and clearly defined clusters.



PART C).

Train Metrics:

MSE: 24189000.18

RMSE: 4918.23

R2 Score: 0.03

Adjusted R2 Score: 0.01

MAE: 3976.71

Test Metrics:

MSE: 24129383.98

RMSE: 4912.17

R2 Score: 0.01

Adjusted R2 Score: -0.07

MAE: 3797.87

PART D).

Comparison of Selected Features Model vs Full Feature Model:

- Mean Squared Error (MSE) and Root Mean Squared Error (RMSE):
    1. Train: Selected model has slightly higher errors
    2. Test: Selected model has lower errors This suggests the selected features model may generalize better to unseen data.
- R2 Score:
    1. Both models have very low R2 scores (0.01-0.03), indicating poor fit.
    2. Selected model performs slightly better on the test set (0.03 vs 0.01).
- Adjusted R2 Score:
    1. Train: Both models perform equally (0.01)
    2. Test: Selected model (0.02) outperforms full model (-0.07) The negative adjusted R2 for the full model on the test set suggests overfitting

- Mean Absolute Error (MAE):
    1. Train: Selected model slightly higher (3982.15 vs 3976.71)
    2. Test: Selected model lower (3772.90 vs 3797.87) Consistent with other metrics, indicating better generalization

Overall Analysis:

- **Generalization:** The selected features model shows better generalization to unseen data, evidenced by improved performance on the test set across all metrics.
- **Overfitting**: The full feature model shows signs of overfitting, particularly with its negative adjusted R2 score on the test set. The selected features model reduces this overfitting.
- **Model Performance**: Both models have very low R2 and adjusted R2 scores, indicating that neither explains much variance in the target variable. This suggests linear regression might not be suitable for this dataset.
- **Consistency:** The selected features model shows more consistent performance between train and test sets, while the full feature model has a larger gap.
- **Simplicity:** The selected features model uses only 3 features, making it simpler and potentially more interpretable.

While the selected features model shows slightly better generalization and less overfitting, the overall performance of both models is poor. The very low R2 scores suggest that linear regression isn't capturing the underlying patterns effectively.