Assignment

Homework-1

Aman kudiyal
2021128

CSE343: Machine Learning Assignment-1

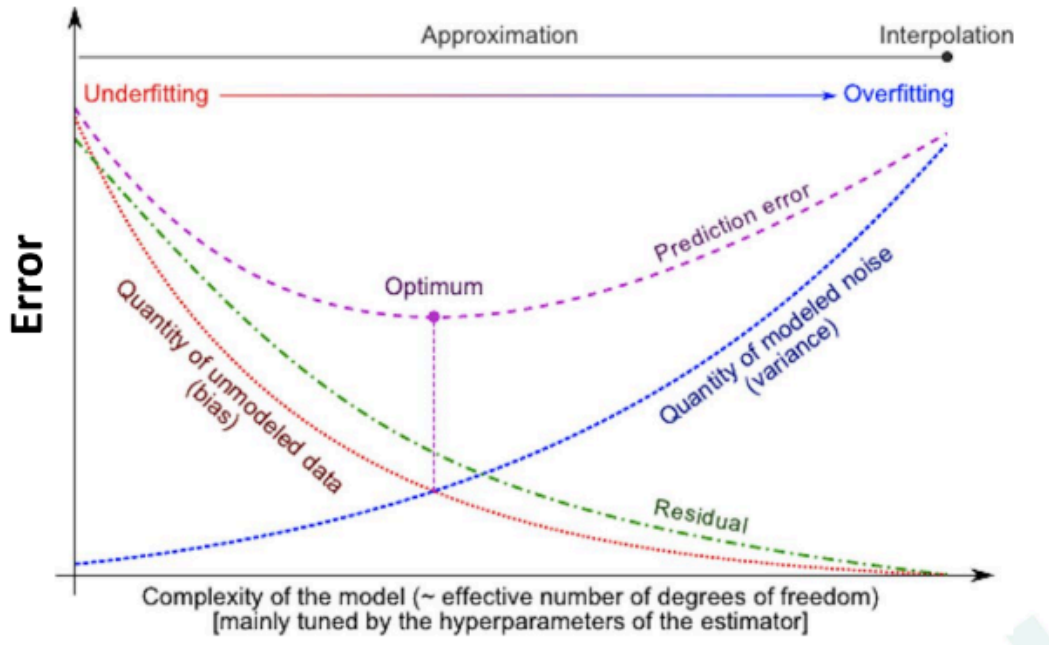
Section-A

A)

When building a machine-learning model , one at which one decides to make it more complex, adding more features or using higher-degree polynomials in a regression, for example—one is essentially giving the model more flexibility to fit the training data. And this is usually what happens:

1. **The Model Fits the Training Data Better**: With increased complexity, the model can capture more intricate patterns in the data. Because it self-tunes to approximate every individual data point, it cuts down on a number of errors related to the simplification of assumptions. That is, bias decreases.
2. **The Model performs poor to new** : However, this extra flexibility means the model starts to overfit the training data. Due to which it performs very poorly to the new incoming data as it becomes less generalized to new data.This leads to an **increase in variance**.

So in summary every time we try to change the complexity of the model, we would need to assess the bias-variance tradeoff for the change.



In the graph of lecture:5 also we can see an intersection between bias and variance graph which is considered as the best bias variance tradeoff.

B)

True Positives (TP): Spam emails correctly identified as spam
False Positives (FP): Legitimate emails incorrectly identified as spam
False Negatives (FN): Spam emails incorrectly identified as legitimate
True Negatives (TN):.Legitimate emails correctly identified as legitimate

TP=200
FP=20
FN=50
TN=730

Precision(spam)= $\frac{TP}{TP+FP}$

=> 200/(200+20)=0.909

Recall(spam)= $\frac{TP}{TP+FN}$

=> 200/(200+50)=0.8

F1 Score(spam)=2× $\frac{Precision(spam)\times Recall(spam)}{PRecision(spam)+recall(spam)}$

=>2× $\frac{0.9091\times 0.80}{0.9091+\ 0.80}$ **=0.8511**

Accuracy= $\frac{TP+TN}{TP+FN+TN+FP}$

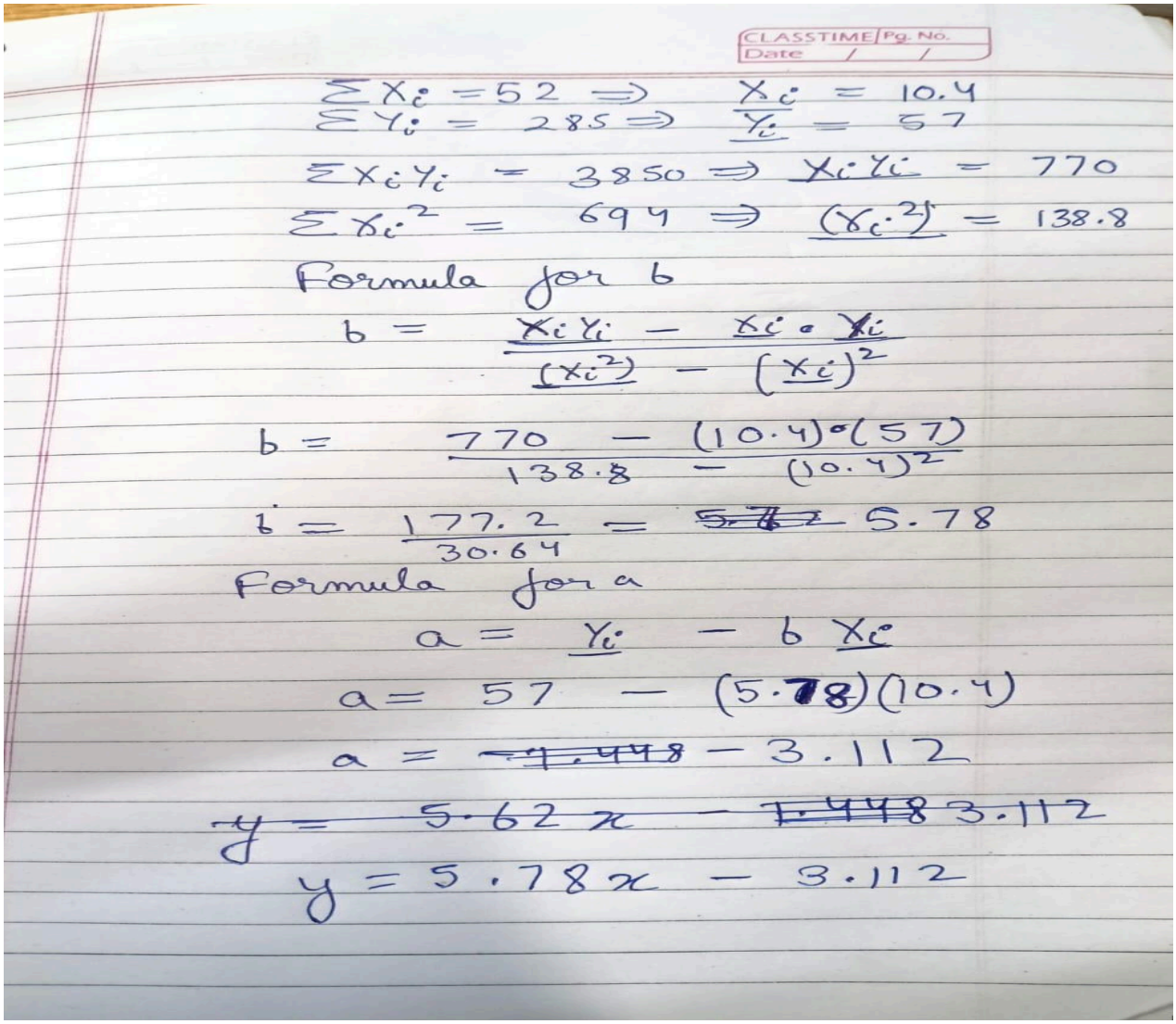
=> $\frac{200+730}{1000}$ **=0.93**

C) To find the equation of the regression line for the given data, we will use the method of least squares for linear regression.

Let the equation be : y=a+bx

Computing the necessary terms:

Xi	Yi	Xi2	Xi.Yi
3	15	9	45
6	30	36	180
10	55	100	550
15	85	225	1275
18	100	324	1800
Total		694	3850



d)

Model f1: A **high-degree polynomial** model (e.g., a 3rd-degree polynomial).

$$f1(x)=ax^4+bx^3+cx^2+dx+e$$

Model f2: A **linear** model (e.g., simple linear regression).

$$f2(x)=mx+c$$

Training Loss (Empirical Risk)

- **Model f1**: The training loss is very low because the model fits the data perfectly. However, this comes at the cost of overfitting, as the model captures both the signal and the noise.
- **Model f2**: The training loss is higher compared to f1, but it captures the general trend without overfitting to the noise.

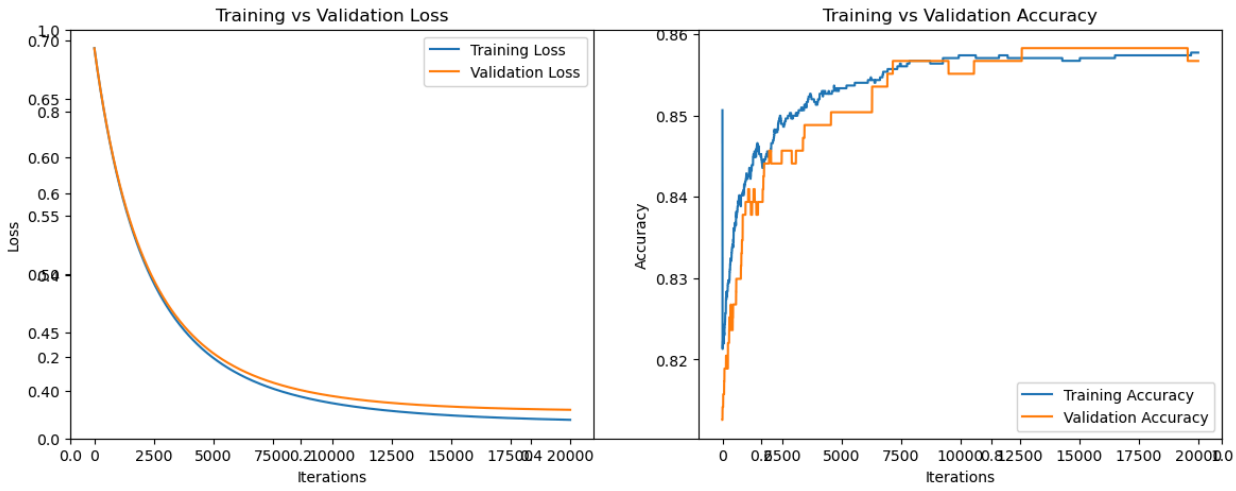
Generalization

- **Model f1**: Despite its low training loss, Model f1 is expected to generalize poorly to new data because it overfits the training data, including noise.
- **Modelf2**: Although it has a higher training loss, Model f2 is more likely to generalize well because it captures the true linear relationship between x and y without being affected by the noise.

This comparison clearly shows the trade-off between model complexity and generalization. While Model f1 performs perfectly on the training data, it overfits and is likely to perform poorly on new data. In contrast, Model f2 doesn't fit as closely but captures the underlying trend, making it more likely to generalize well. Choosing the right model complexity is crucial to balance fitting the data and generalizing for future predictions.

Section-B

a)



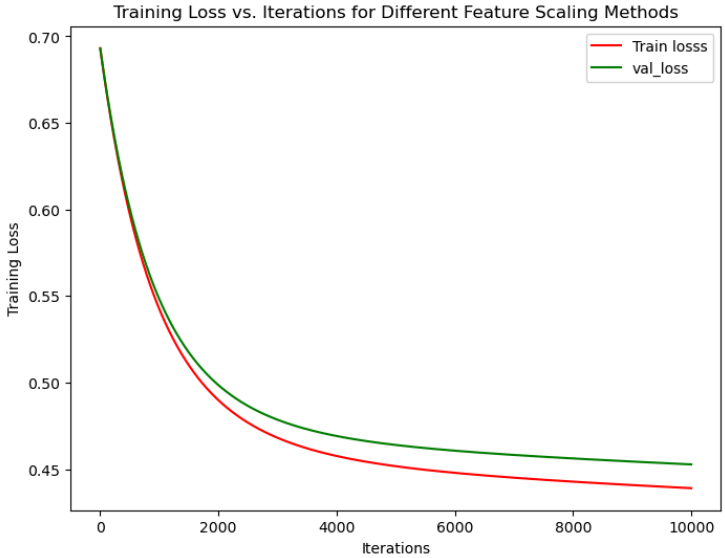
Convergence of Loss: Both the training and validation loss decrease sharply at the beginning, indicating that the model is learning effectively from the data. They then stabilize, showing that the model is approaching an optimal state.

The gap between the training and validation loss is small and consistent. This indicates that the model is not overfitting significantly, as it generalizes well to unseen data.

Convergence of Accuracy: The training accuracy increases sharply initially and then levels off, which is typical in training processes as the model begins to fit the data well.

Validation Accuracy: The validation accuracy increases in steps rather than smoothly. And is greater than training accuracy momentarily

b)



No Scaling: The training loss starts very high and experiences significant fluctuations throughout the iterations thus not having a stable loss also after 10k iterations. Similarly the validation loss acts like training loss

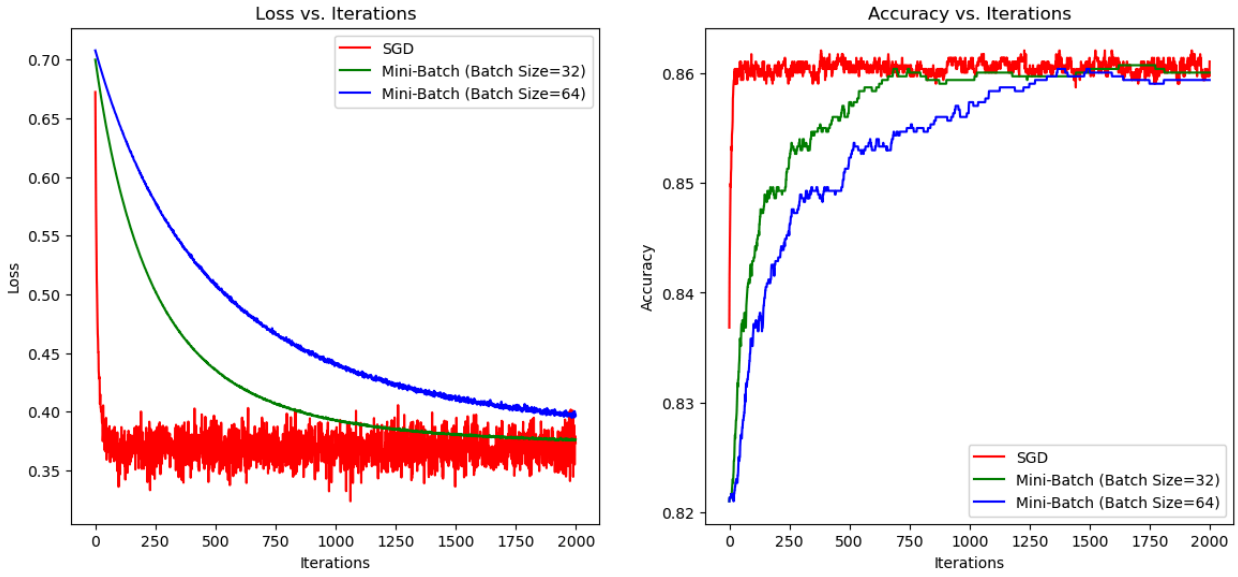
Min-Max Scaling: With Min-Max scaling, both training and validation losses start at a lower point and smoothly decrease, converging more consistently

c)

```
Confusion Matrix:
[[520  22]
 [ 80  13]]
Precision: 0.37
Recall: 0.14
F1 Score: 0.20
ROC-AUC Score: 0.69
```

Precision is low, meaning lots of false positives. There is also low recall, with large numbers of false negatives. The F1 score is similarly low, indicating a poor balance between precision and recall. From these metrics, it appears that the model is having a hard time accurately classifying the positive class and probably needs improvements or consideration with respect to the feature used. In addition, the dataset was somewhat problematic.But here a 0.69 ROC-AUC suggests that the model is fairly capable of distinguishing between the positive and negative classes

d)



SGD improves accuracy quickly but is highly variable, with noisy updates and fluctuations. A mini-batch of 32 offers faster, more stable progress than SGD, striking a good balance between speed and stability. A batch size of 64 improves slower but with the greatest stability and least fluctuation.

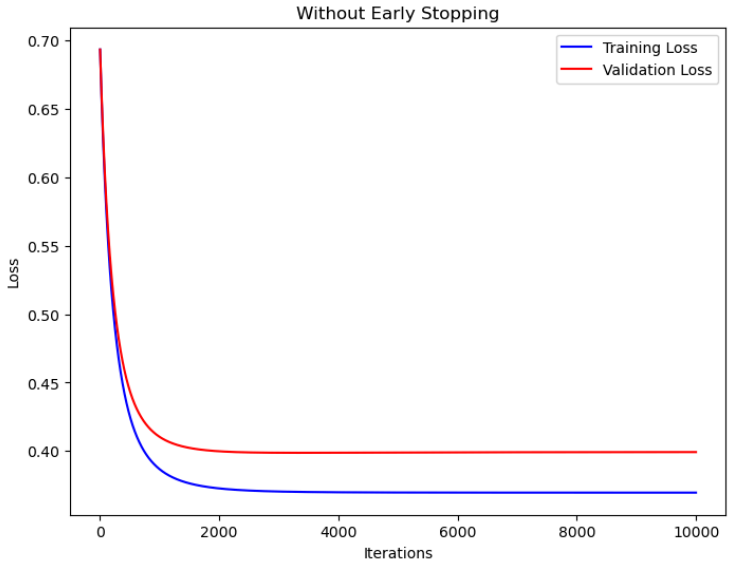
SGD is fastest but most volatile, while smaller mini-batches converge quickly with less stability. Larger mini-batches are slower but provide more consistent training. The choice depends on the need for speed or stability and available resources.

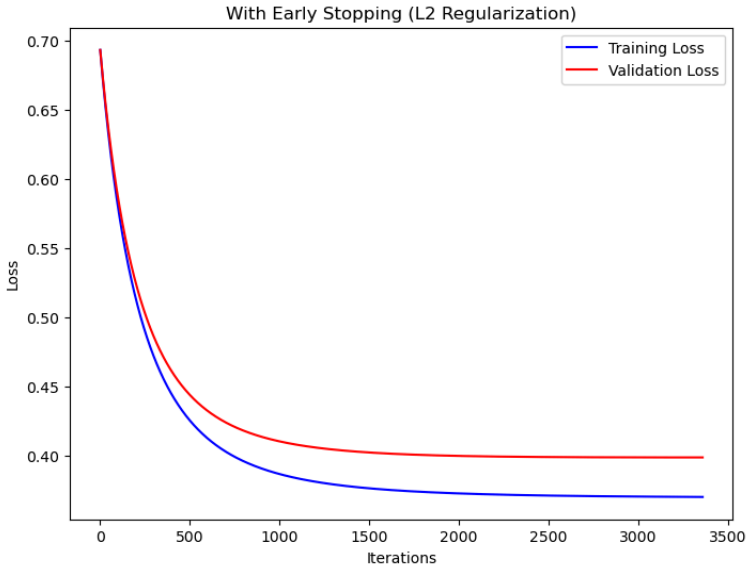
e)

```
Average Metrics Across Folds:
Accuracy: 0.8535 ± 0.0048
Precision: 0.7133 ± 0.1343
Recall: 0.0621 ± 0.0109
F1 Score: 0.1140 ± 0.0191
```

The mean accuracy is quite high, showing that the model performs well overall. Low standard deviation indicates stable performance across folds. However, precision varies significantly due to high variability in the data used in each fold. The average recall is quite low, which means the model is missing a large number of relevant cases (true positives) across folds, and the F1 score is also low, confirming struggles with balancing precision and recall. While accuracy is stable, recall and F1 remain low

f)



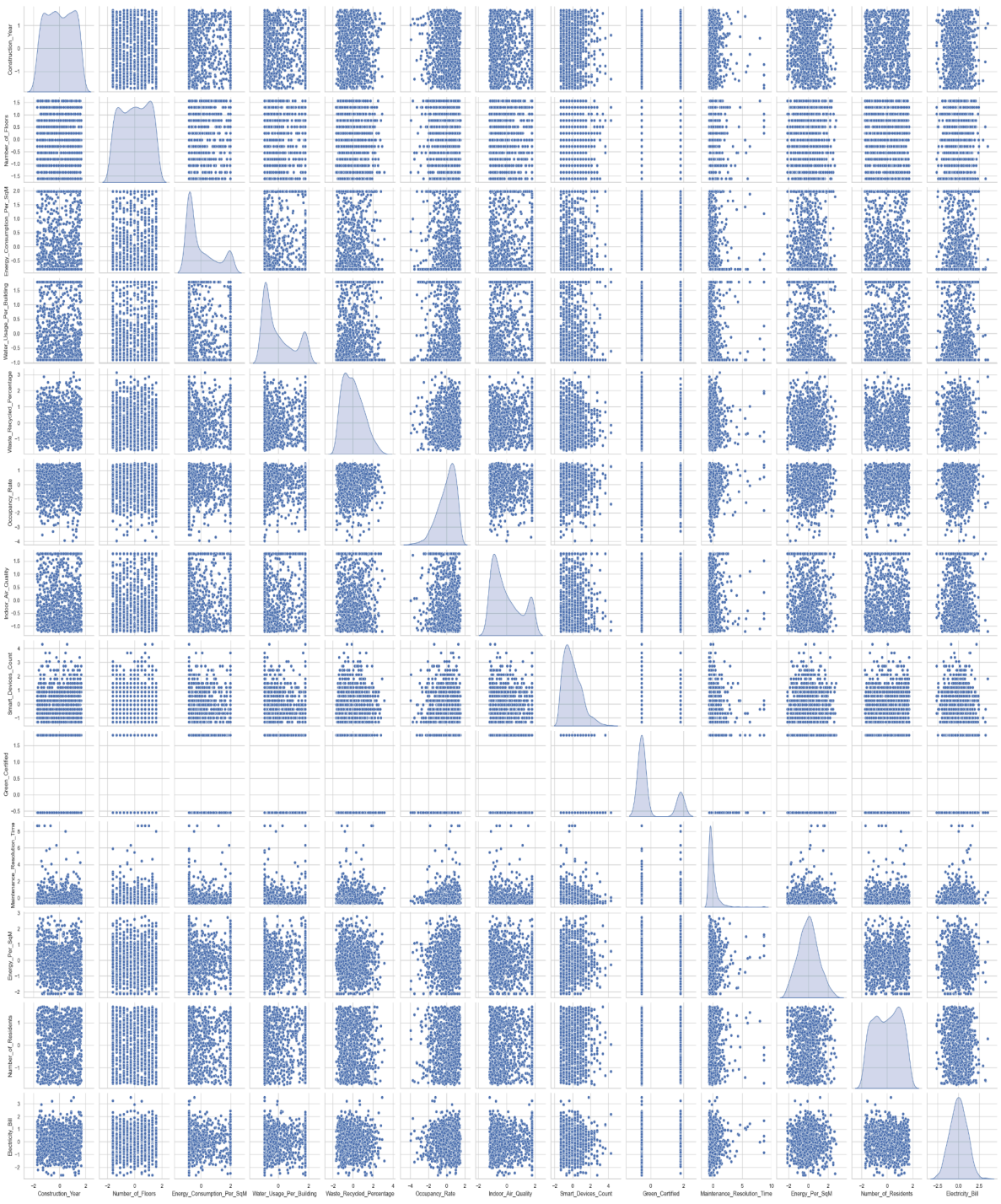


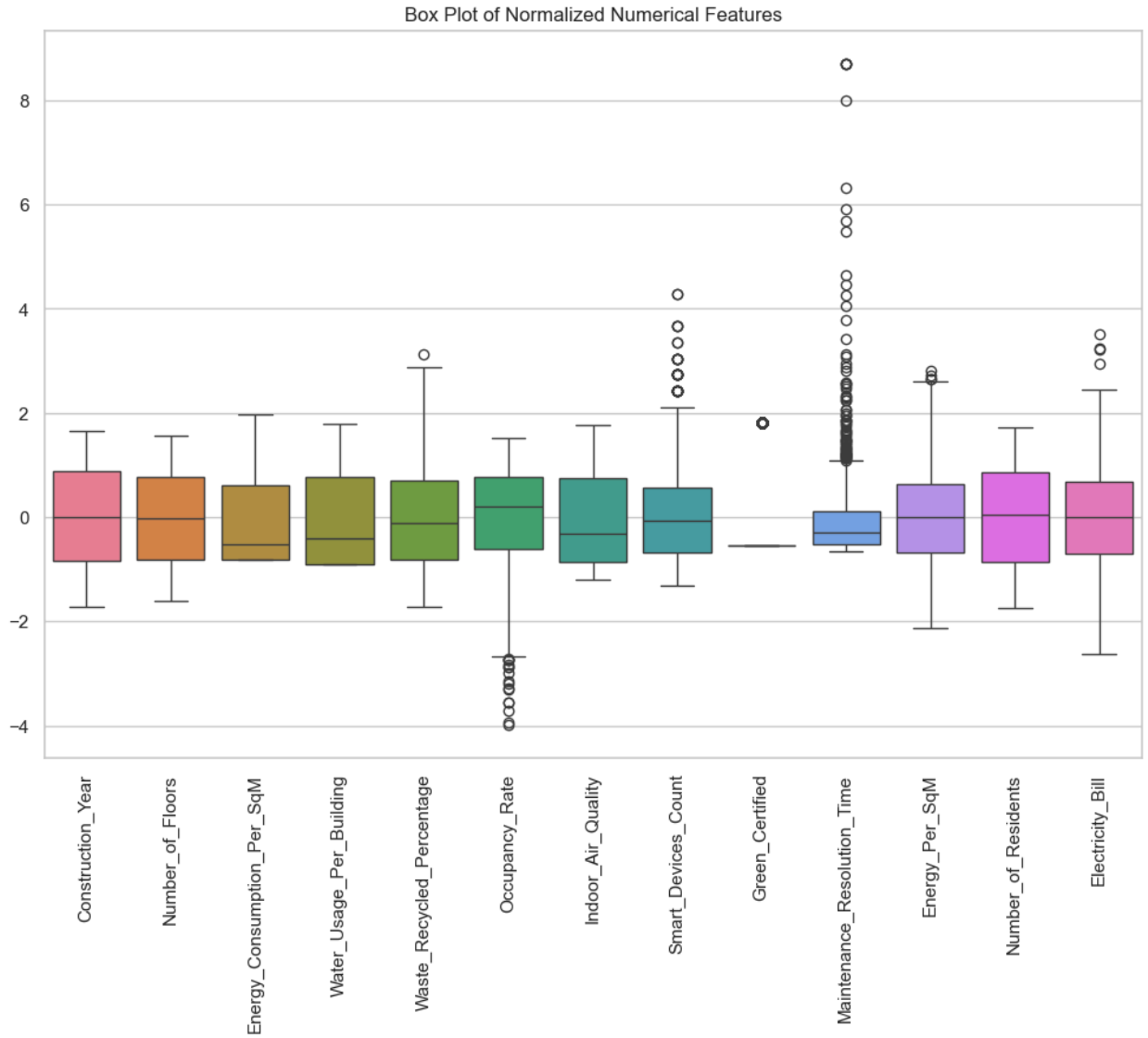
Since in the above accuracy graphs the highest accuracy is observed in batch gradient decent so I applied early stopping in batch gradient descent.In this we can see that the early stopping helps us to avoid overfitting and also helps us to reduce the total training iterations as it breaks from gradient descent in between.

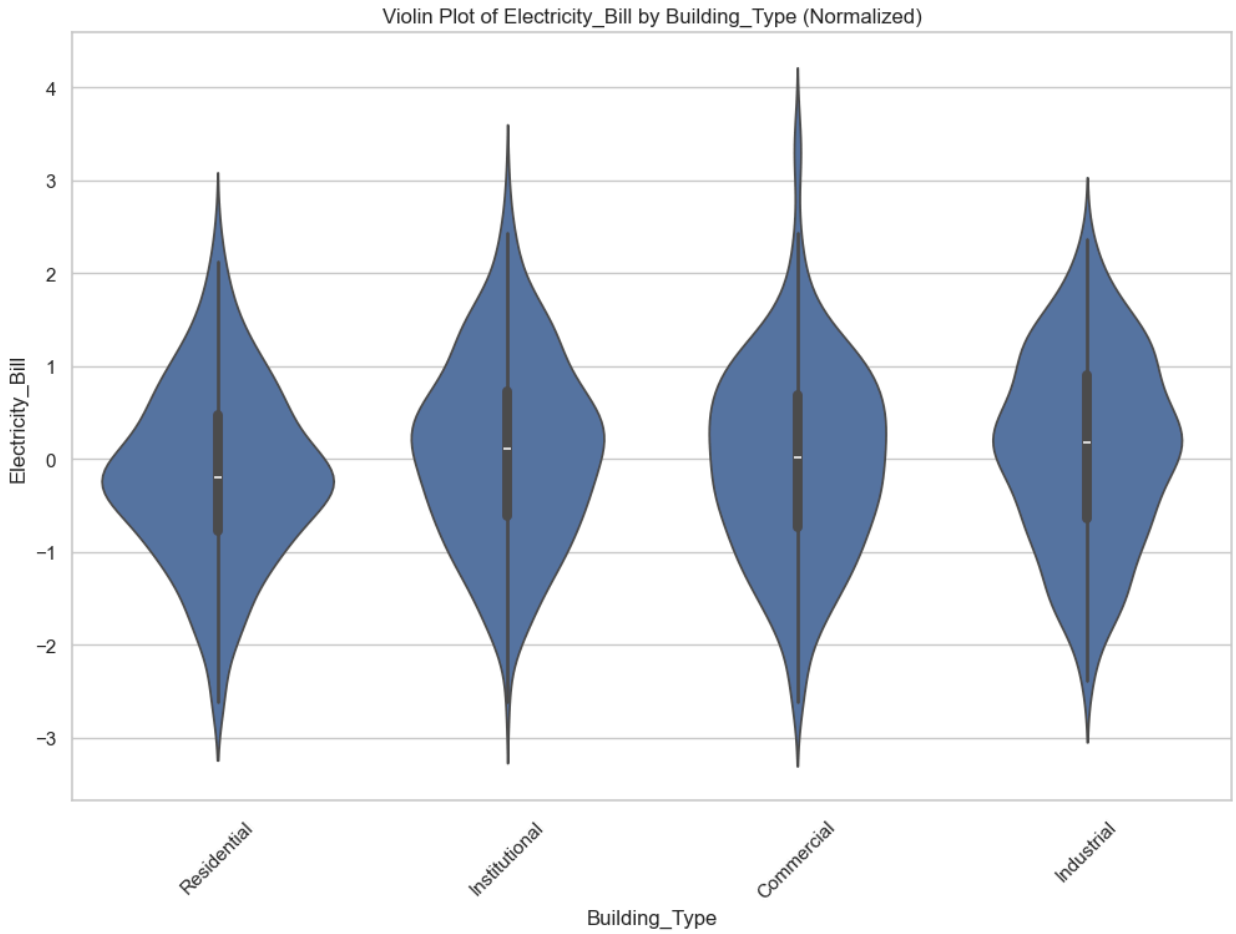
Section-C (bonus)

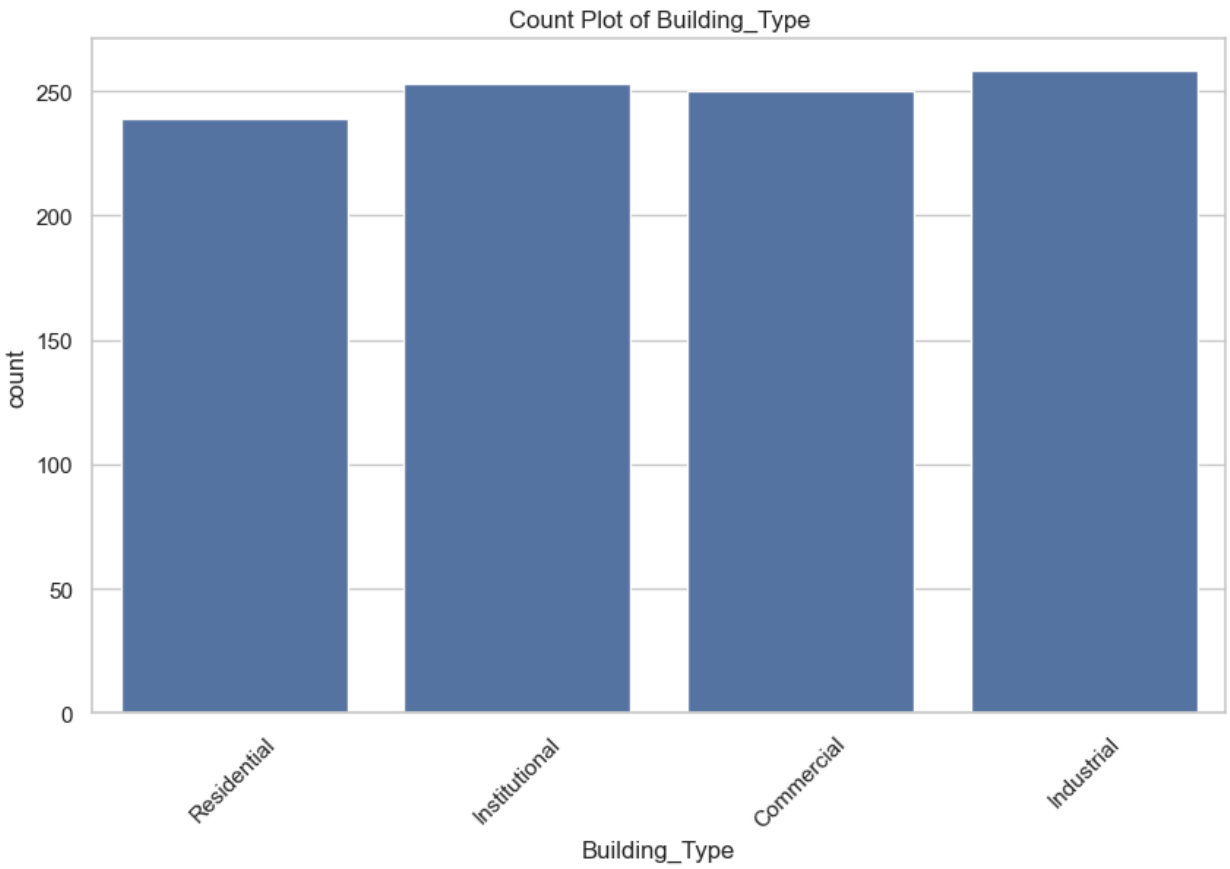
a)

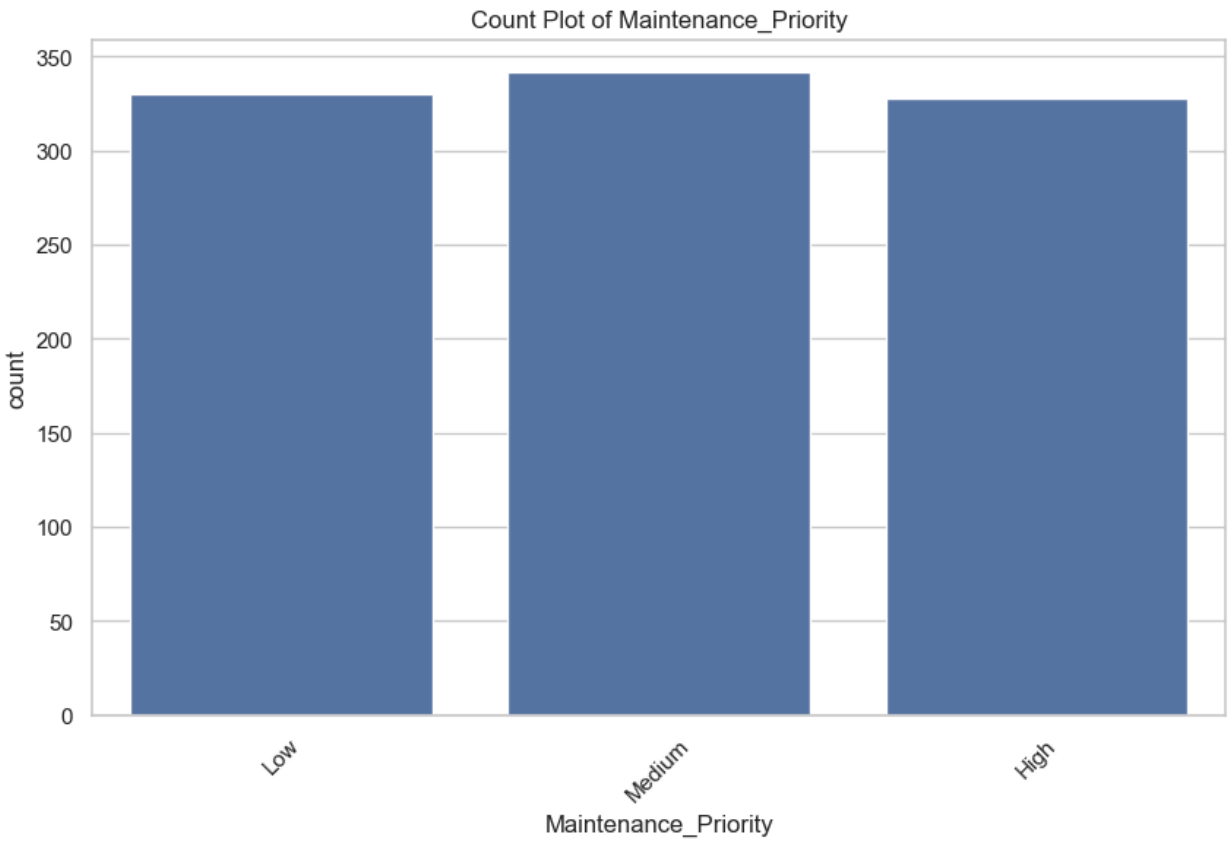
Pair Plot of Normalized Numerical Features

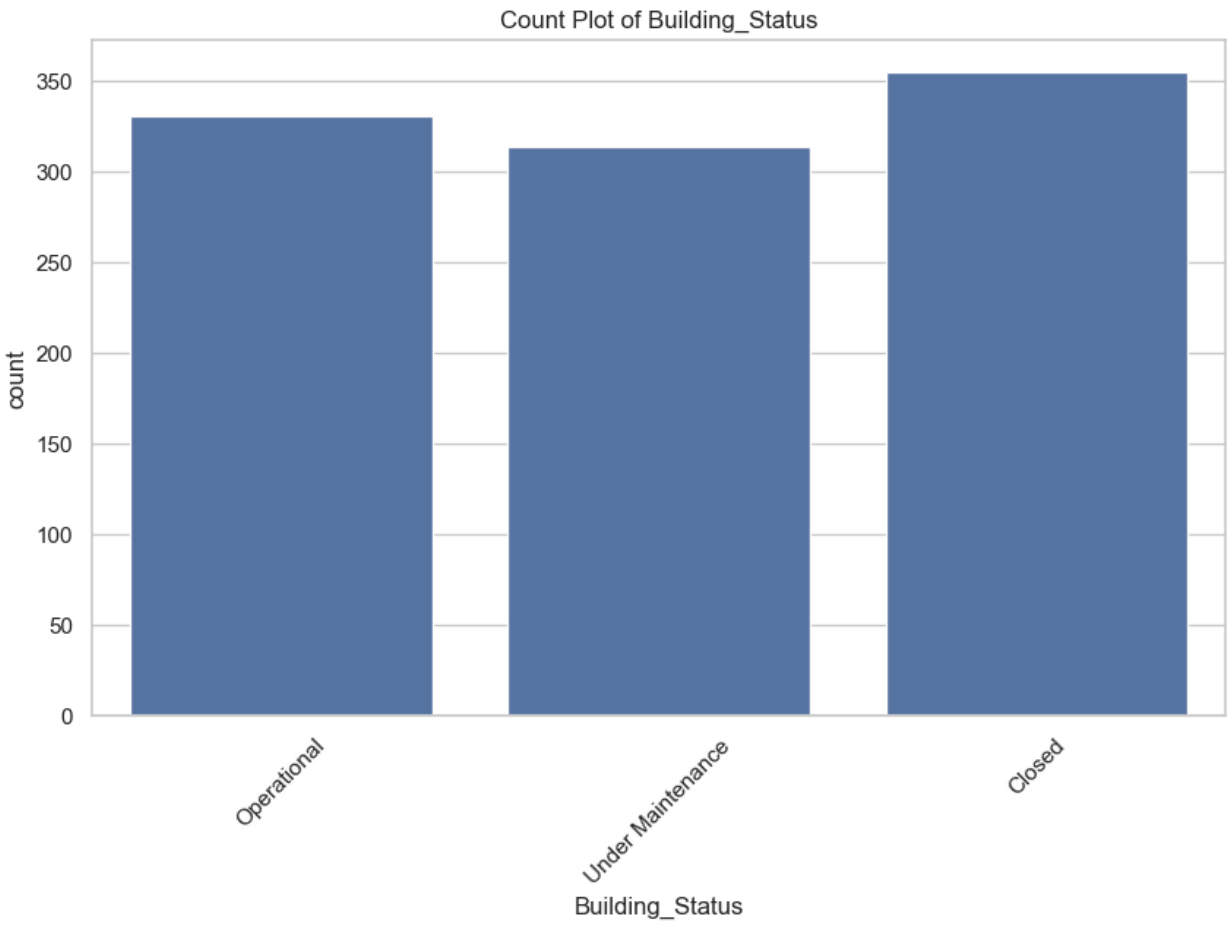


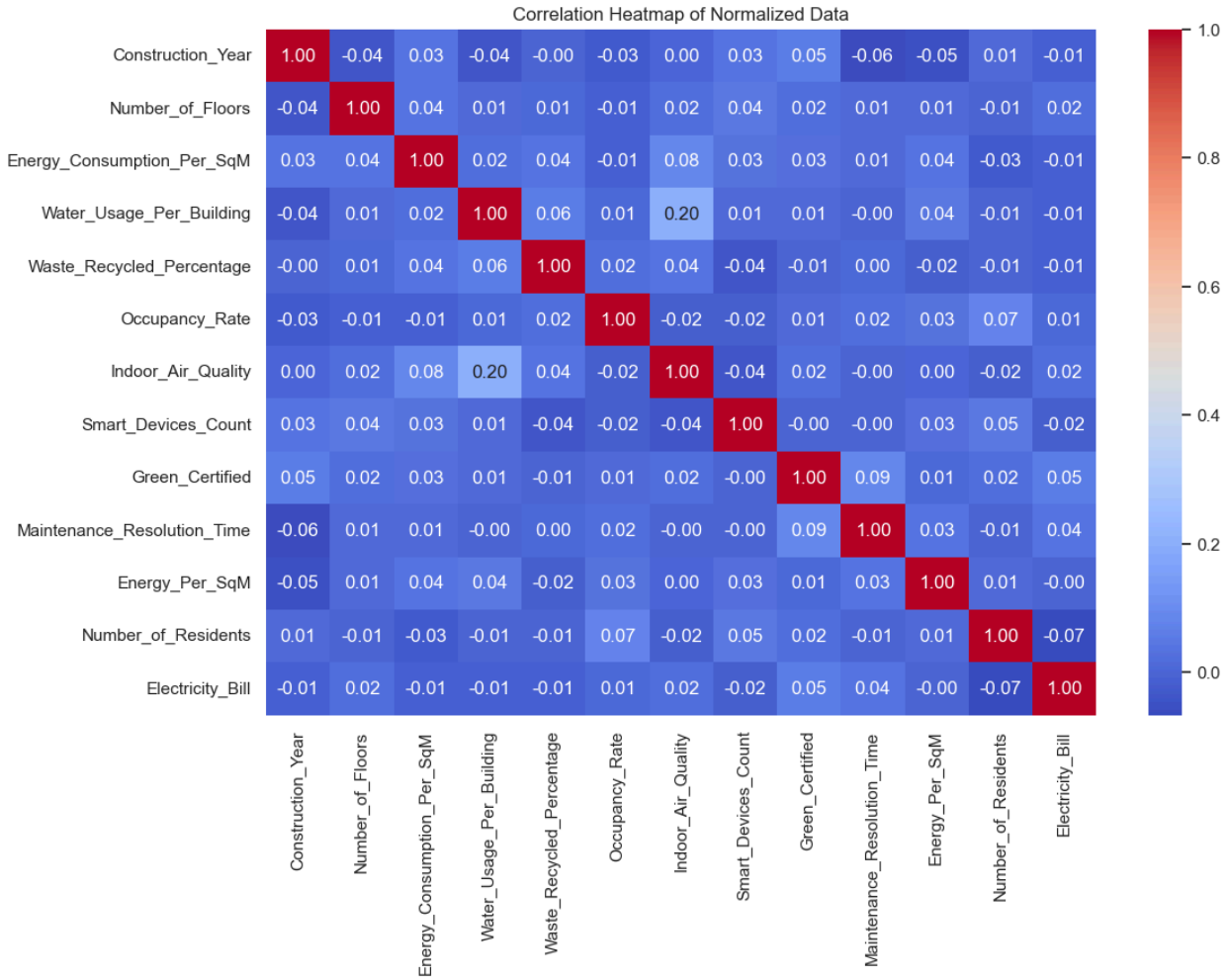

Box Plot of Normalized Numerical Features
8
6
4
2
0
−2
−4
Construction_Year
Number_of_Floors
Energy_Consumption_Per_SqM
Water_Usage_Per_Building
Waste_Recycled_Percentage
Occupancy_Rate
Indoor_Air_Quality
Smart_Devices_Count
Green_Certified
Maintenance_Resolution_Time
Energy_Per_SqM
Number_of_Residents
Electricity_Bill


Violin Plot of Electricity_Bill by Building_Type (Normalized)
Electricity_Bill
4
3
2
1
0
−1
−2
−3
Residential
Institutional
Commercial
Industrial
Building_Type


Count Plot of Building_Type
count
0
50
100
150
200
250
Residential
Institutional
Commercial
Industrial
Building_Type


Count Plot of Maintenance_Priority
count
350
300
250
200
150
100
50
0
Low
Medium
High
Maintenance_Priority


Count Plot of Building_Status
count
0
50
100
150
200
250
300
350
Operational
Under Maintenance
Closed
Building_Status



1)In the correlation graph we can see that almost every corelation is very small

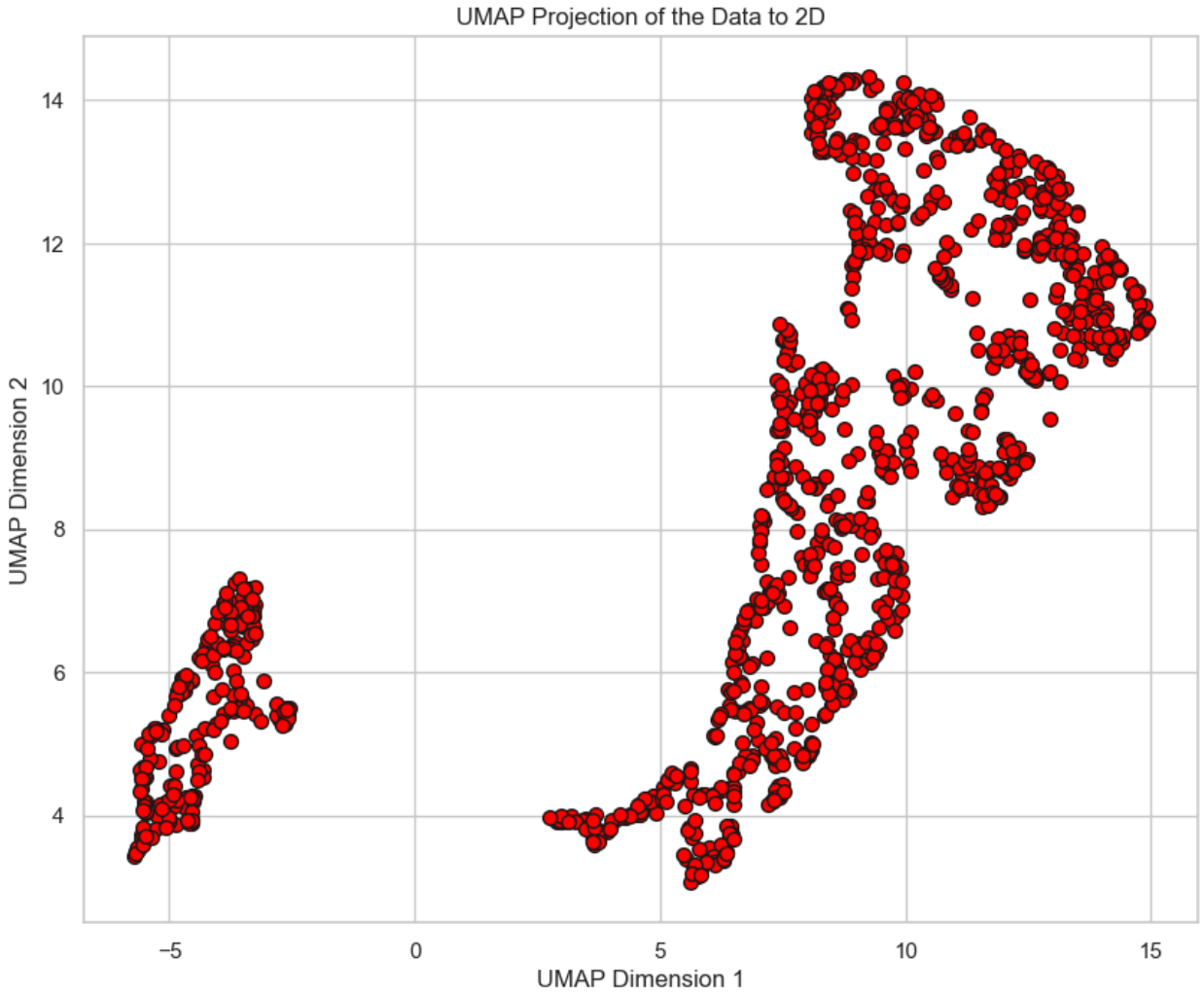
2) But there is a moderate correlation (0.20) between(Indoor Air Quality and Water Usage per building) . This could suggest that buildings with more water usage might be better in monitoring air quality effectively.

3) The violin plots comparing Electricity_Bill across different Building_Types reveal some interesting insights. Industrial buildings, in particular, stand out with a broader range of electricity bills and a potentially higher median. This could suggest that these buildings consume more energy or have unique usage patterns compared to others.

4)In data, we found that the (Smart_Devices_Count) and (Maintenance_Resolution_Time) have have a significant number of outliers. This could impact model performance if not addressed.

5)The count plot of Building_Type shows a fairly uniform distribution across different types of buildings (Residential, Institutional, Commercial, Industrial), which is good for model training as it provides a balanced view across different categories.

B)



The UMAP projection reveals clear, well-defined clusters.The separation between these groups suggests that models could more easily differentiate them, potentially leading to better classification or prediction results. These clusters likely represent hidden patterns or natural groupings within the dataset that aren't immediately visible in the original, more complex space. Overall, this distinct clustering is a positive indication of the dataset's structure and its suitability for machine learning tasks.

C)

```
Columns in the dataset: Index(['Building_Type', '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',
       'Building_Status', 'Maintenance_Priority', 'Energy_Per_SqM',
       'Number_of_Residents', 'Electricity_Bill'],
      dtype='object')
'Electricity_Bill' column found, proceeding with preprocessing.
Train Metrics:
MSE: 24475013.1685, RMSE: 4947.2228, MAE: 4006.3285, R2: 0.0139, Adjusted R2: -0.0011

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

Overall, these metrics suggest that your model may not be capturing the underlying patterns of the data effectively, leading to poor performance both in training and testing. This could be due to several factors like underfitting, where the model is too simple to capture the complex patterns in the data, or possibly issues with data quality .We might need checking for data issues, or trying different algorithms.

D)

```
Top 3 selected features using RFE: Index(['Building_Type', 'Green_Certified', 'Number_of_Residents'], dtype='object')

Metrics with All Features (Previous Results):
Train - MSE: 24475013.1685, RMSE: 4947.2228, MAE: 4006.3285, R2: 0.0139, Adjusted R2: -0.0011
Test - MSE: 24278016.1557, RMSE: 4927.2727, MAE: 3842.4093, R2: 0.0000, Adjusted R2: -0.0641

Metrics with Selected Features (RFE):
Train - MSE: 24569032.9069, RMSE: 4956.7159, MAE: 4006.4734, R2: 0.0101, Adjusted R2: 0.0072
Test - MSE: 23941409.0630, RMSE: 4892.9959, MAE: 3813.9481, R2: 0.0139, Adjusted R2: 0.0019
```

The model with selected features (RFE) generally shows better or comparable performance on the test set across most metrics except for MSE and RMSE, where the differences are slight. This suggests a trade-off where simplifying the model by using fewer features does not substantially degrade performance and in fact, slightly improves generalization as seen in MAE and adjusted R^2.

E)

```
Metrics with Ridge Regression (One-Hot Encoded):
Train - MSE: 24475013.5178, RMSE: 4947.2228, MAE: 4006.3092, R2: 0.0139, Adjusted R2: -0.0011
Test - MSE: 24277790.8280, RMSE: 4927.2498, MAE: 3842.3729, R2: 0.0000, Adjusted R2: -0.0641

Metrics with All Features (Linear Regression - Previous Results):
Train - MSE: 24475013.1685, RMSE: 4947.2228, MAE: 4006.3285, R2: 0.0139, Adjusted R2: -0.0011
Test - MSE: 24278016.1557, RMSE: 4927.2727, MAE: 3842.4093, R2: 0.0000, Adjusted R2: -0.0641
```

The comparison shows that switching from Linear Regression with all features to Ridge Regression with one-hot encoded features doesn't significantly change the model's performance in terms of error metrics and the ability to explain the variance in the target. Both models struggle similarly, implying that the primary issues might lie in feature selection, the inherent noise within the dataset, or perhaps the linear assumptions of the models themselves.

F)

```
ICA with 4 components:
Train - MSE: 24741916.8861, RMSE: 4974.1247, MAE: 4004.2797, R2: 0.0032, Adjusted R2: -0.0008
Test - MSE: 24278454.4126, RMSE: 4927.3172, MAE: 3839.1392, R2: 0.0000, Adjusted R2: -0.0163

ICA with 5 components:
Train - MSE: 24733570.3041, RMSE: 4973.2857, MAE: 4001.6326, R2: 0.0035, Adjusted R2: -0.0015
Test - MSE: 24245098.5114, RMSE: 4923.9312, MAE: 3835.3360, R2: 0.0014, Adjusted R2: -0.0191

ICA with 6 components:
Train - MSE: 24634379.4475, RMSE: 4963.3033, MAE: 4009.0204, R2: 0.0075, Adjusted R2: 0.0015
Test - MSE: 24476361.7086, RMSE: 4947.3591, MAE: 3844.7836, R2: -0.0081, Adjusted R2: -0.0330

ICA with 8 components:
Train - MSE: 24629118.5543, RMSE: 4962.7733, MAE: 4008.7540, R2: 0.0077, Adjusted R2: -0.0003
Test - MSE: 24471866.5222, RMSE: 4946.9047, MAE: 3844.2824, R2: -0.0079, Adjusted R2: -0.0414
```

The ICA component analysis reveals that increasing the number of components marginally improves training performance but does not consistently benefit test performance, potentially leading to models that do not generalize well outside of the training data. The best compromise appears to be using around 5 components, which offers the lowest test MSE and RMSE, suggesting a better balance between complexity and predictive performance.

G)

```
ElasticNet with alpha=0.01:
Train - MSE: 24475021.9227, RMSE: 4947.2237, MAE: 4006.2321, R2: 0.0139, Adjusted R2: -0.0011
Test - MSE: 24276881.6860, RMSE: 4927.1576, MAE: 3842.2260, R2: 0.0001, Adjusted R2: -0.0640

ElasticNet with alpha=0.1:
Train - MSE: 24475813.0835, RMSE: 4947.3036, MAE: 4005.4693, R2: 0.0139, Adjusted R2: -0.0011
Test - MSE: 24267836.2170, RMSE: 4926.2396, MAE: 3841.1333, R2: 0.0005, Adjusted R2: -0.0636

ElasticNet with alpha=0.5:
Train - MSE: 24489015.1753, RMSE: 4948.6377, MAE: 4003.2022, R2: 0.0134, Adjusted R2: -0.0017
Test - MSE: 24245264.5104, RMSE: 4923.9481, MAE: 3837.7549, R2: 0.0014, Adjusted R2: -0.0626

ElasticNet with alpha=1.0:
Train - MSE: 24513721.6910, RMSE: 4951.1334, MAE: 4002.0648, R2: 0.0124, Adjusted R2: -0.0027
Test - MSE: 24237702.1967, RMSE: 4923.1801, MAE: 3835.1277, R2: 0.0017, Adjusted R2: -0.0623

ElasticNet with alpha=5.0:
Train - MSE: 24652049.1018, RMSE: 4965.0830, MAE: 4003.2930, R2: 0.0068, Adjusted R2: -0.0084
Test - MSE: 24277056.6274, RMSE: 4927.1753, MAE: 3834.6443, R2: 0.0001, Adjusted R2: -0.0640

ElasticNet with alpha=10.0:
Train - MSE: 24716377.1929, RMSE: 4971.5568, MAE: 4005.0301, R2: 0.0042, Adjusted R2: -0.0110
Test - MSE: 24307831.3886, RMSE: 4930.2973, MAE: 3837.6452, R2: -0.0012, Adjusted R2: -0.0654
```

The ElasticNet model performs optimally at an alpha value of 1.0 based on the test metrics, providing the best balance between error minimization and the complexity of the model. Both lower and higher values of alpha result in poorer performance, indicating that the correct level of regularization is crucial for achieving optimal predictive accuracy.

H)

```
Metrics with Gradient Boosting Regressor:
Train - MSE: 14926446.2573, RMSE: 3863.4759, MAE: 3092.7482, R2: 0.3986, Adjusted R2: 0.3895
Test - MSE: 24392500.9011, RMSE: 4938.8765, MAE: 3815.7032, R2: -0.0047, Adjusted R2: -0.0691

Metrics with All Features (Linear Regression - Part c):
Train - MSE: 24475013.1685, RMSE: 4947.2228, MAE: 4006.3285, R2: 0.0139, Adjusted R2: -0.0011
Test - MSE: 24278016.1557, RMSE: 4927.2727, MAE: 3842.4093, R2: 0.0000, Adjusted R2: -0.0641

Metrics with ElasticNet Regularization - Part g:

Alpha: 0.01
Test - MSE: 24276881.6860, RMSE: 4927.1576, MAE: 3842.2260, R2: 0.0001, Adjusted R2: -0.0640

Alpha: 0.1
Test - MSE: 24267836.2170, RMSE: 4926.2396, MAE: 3841.1333, R2: 0.0005, Adjusted R2: -0.0636

Alpha: 0.5
Test - MSE: 24245264.5104, RMSE: 4923.9481, MAE: 3837.7549, R2: 0.0014, Adjusted R2: -0.0626

Alpha: 1.0
Test - MSE: 24237702.1967, RMSE: 4923.1801, MAE: 3835.1277, R2: 0.0017, Adjusted R2: -0.0623

Alpha: 5.0
Test - MSE: 24277056.6274, RMSE: 4927.1753, MAE: 3834.6443, R2: 0.0001, Adjusted R2: -0.0640

Alpha: 10.0
Test - MSE: 24307831.3886, RMSE: 4930.2973, MAE: 3837.6452, R2: -0.0012, Adjusted R2: -0.0654
```

ElasticNet with alpha=1.0 provides a balance between mse,rmse and controlling for overfitting,ALso it has comparably high Adjusted R^2 making it the best choice among the options analyzed for this dataset.

Gradient Boosting, while having the lowest MAE, shows potential overfitting as indicated by its test MSE, RMSE, and very negative test Adjusted R^2.

Linear Regression appears to be the least effective model, with high error metrics but has the low test favorable Adjusted R^2.

Therefore, ElasticNet could potentially enhance model performance by addressing issues related to overfitting and irrelevant features, particularly in complex datasets like this one.