CSE343/ECE343: Machine Learning
Assignment-1 Linear and Logistic Regression, ML in Practice, Empirical Risk Minimization
Max Marks: 25 (Programming: 15, Theory: 10) Due Date: 13/09/2024, 11:59 PM

Instructions

- **Keep collaborations at high-level discussions. Copying/Plagiarism will be dealt with strictly.**
- **Late submission penalty: As per course policy.**
- **Your submission should be a single zip file 2020xxx_HW1.zip (Where *2020xxx* is your roll number). Include all the files (code and report with theory questions) arranged with proper names. A single .pdf report explaining your codes with results, relevant graphs, visualization and solution to theory questions should be there. The structure of submission should follow:**

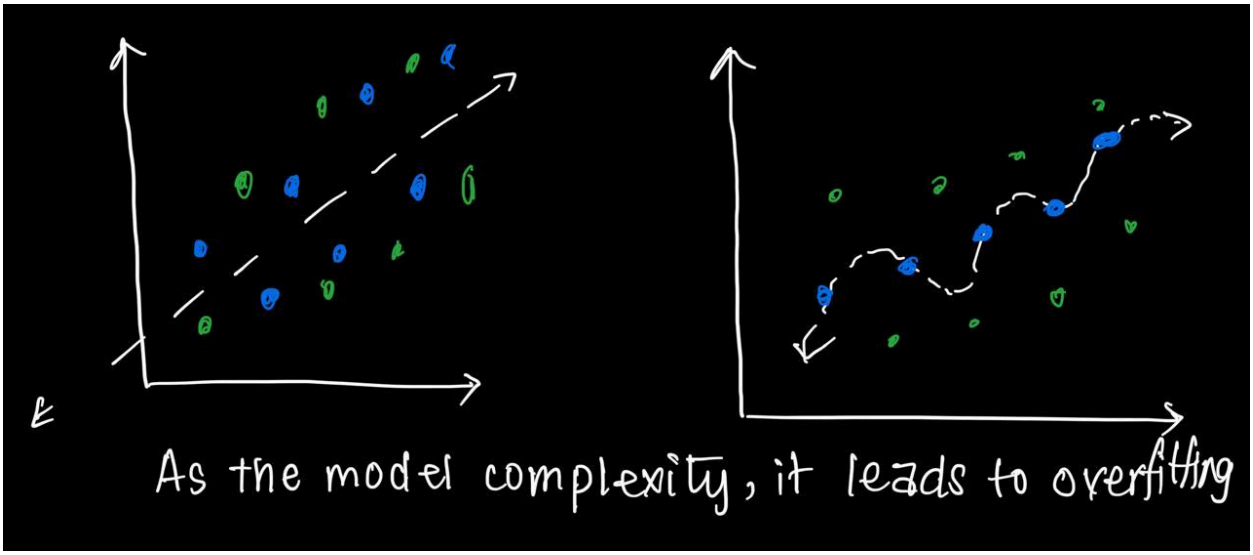
  2020xxx_HW1
  |− code_rollno.py/.ipynb
  |− report_rollno.pdf
  |− (All other files for submission)

- **Anything not in the report will not be graded.**
- **Remember to turn in after uploading on Google Classroom. No excuses or issues would be taken regarding this after the deadline.**
- **Start the assignment early. Resolve all your doubts from TAs in their office hours at least two days before the deadline.**
- **Your code should be neat and well-commented.**
- **You have to do either Section B or C.**
- **Section A is mandatory.**

1. (10 points) Section A (Theoretical)

(a) (2 marks) 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.

Overfitting
As you increase the complexity of the model, it performs well on the training set, leading to low bias but it doesn't generalize well and performs poorly on the testing set leading to bad variance.



(b) (3 marks) 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.

True Positives = 730

True Negatives = 200

False Negatives = 20

False Positives = 50

Average Accuracy = ((730/750) + (200/250))/2 = 0.887 or 88.7%

(c) (3 marks) 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

x	y
3	15
6	30
10	55
15	85
18	100

Table 1: Table of x and y values

equation obtained, predict the value of y when x=12.

x	y
3	15
6	30
10	55
15	85
18	100

$$W = (X^T X)^{-1} X^T Y$$

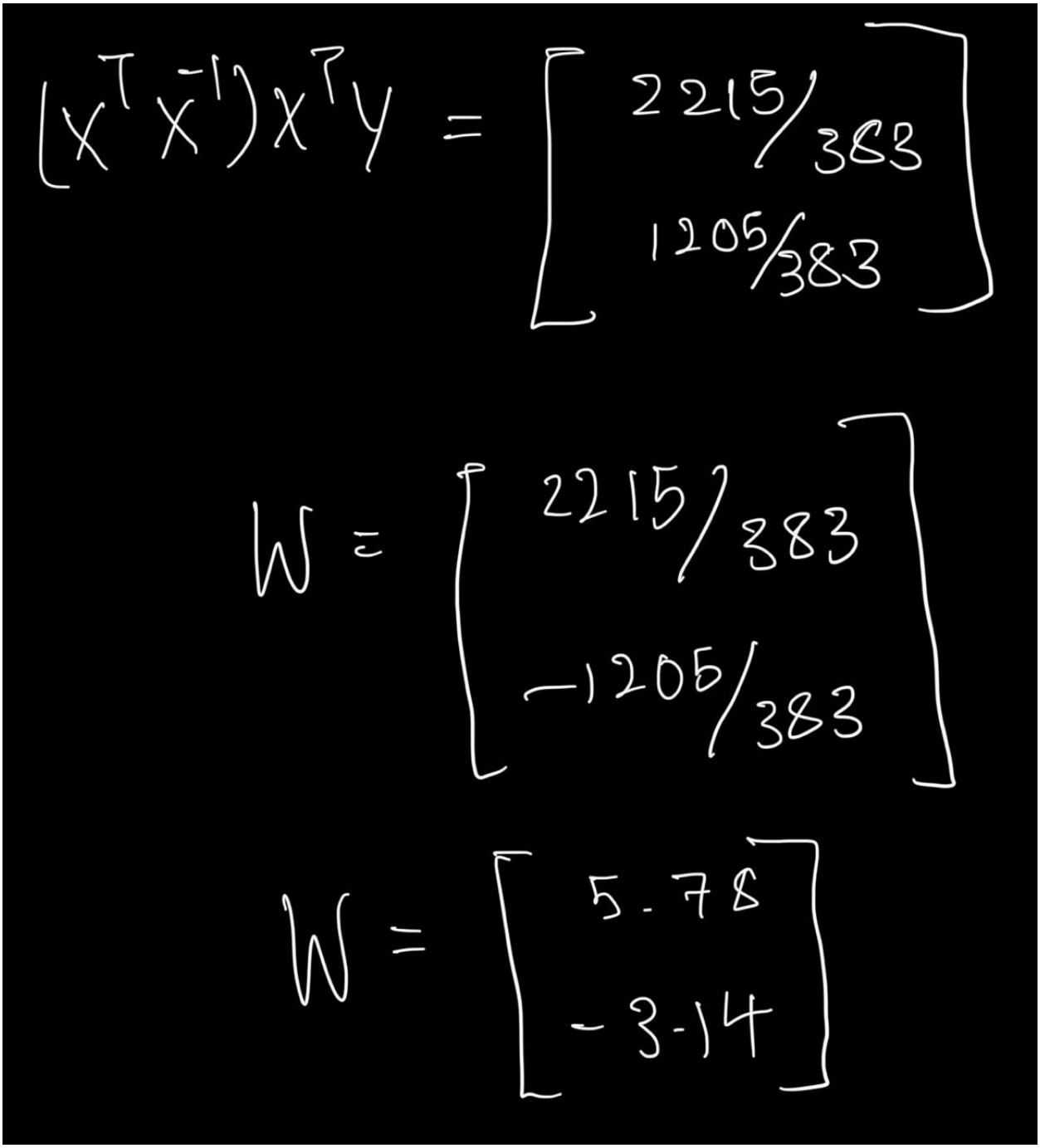
$$X^T X = \begin{bmatrix} 3 & 6 & 10 & 15 & 18 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 6 & 1 \\ 10 & 1 \\ 15 & 1 \\ 18 & 1 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 694 & 52 \\ 52 & 5 \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} 5/766 & -26/383 \\ -26/383 & 347/383 \end{bmatrix}$$

$$(X^T X)^{-1} X^T = \begin{bmatrix} 5/766 & -26/383 \\ -26/383 & 347/383 \end{bmatrix} \begin{bmatrix} 3 & 6 & 10 & 15 & 18 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -37/766 & -11/383 & -1/383 & 23/766 & 19/383 \\ 269/383 & 191/383 & 87/383 & -43/383 & -14/383 \end{bmatrix}$$



The regression line is y = w1 *x + w0 i.e **y = 5.76x - 3.14**

For x =12, y =66.25

(d) (2 marks) Given a training dataset with features X and labels Y, let $\hat{f}(X)$ be the prediction of a model f and $L(\hat{f}(X), Y)$ be the loss function. Suppose you have two models, f_1 and f_2, and the empirical risk for f_1 is lower than that for f_2. Provide a toy example where model f_1 has a lower empirical risk on the training set but may not necessarily generalize better than model f_2.

The toy example would be a good example of overfitting.

In the case of overfitting, the performance is very good on the training dataset but poor on the unseen test dataset.

X	Y
1	2
2	5
3	10

Now

↳ Bias for f_1 should be the least ; $f_1(x) = x^2 + 1$

We see that our emperical risk $R(f_1) = 0$, as the function fits the training data.

For f_2, let $f_2(x) = 2x$.

$$R(f_2) = \frac{(1-2)^2 + (4-4)^2 + (6-10)^2}{3} = 6$$

X	Y
9	19
4	11
5	13

For the f1 function,

$$R(f_1) = \frac{(82-19)^2 + (17-11)^2 + (26-15)^2}{3}$$

$$R(f_1) =$$

For the f2 function,

$$R(f_2) = \frac{(18-19)^2 + (9-11)^2 + (10-13)^2}{3}$$

$$R(f_2) = 4.67$$

$R(f_2) < R(f_1)$

This shows overfitting.

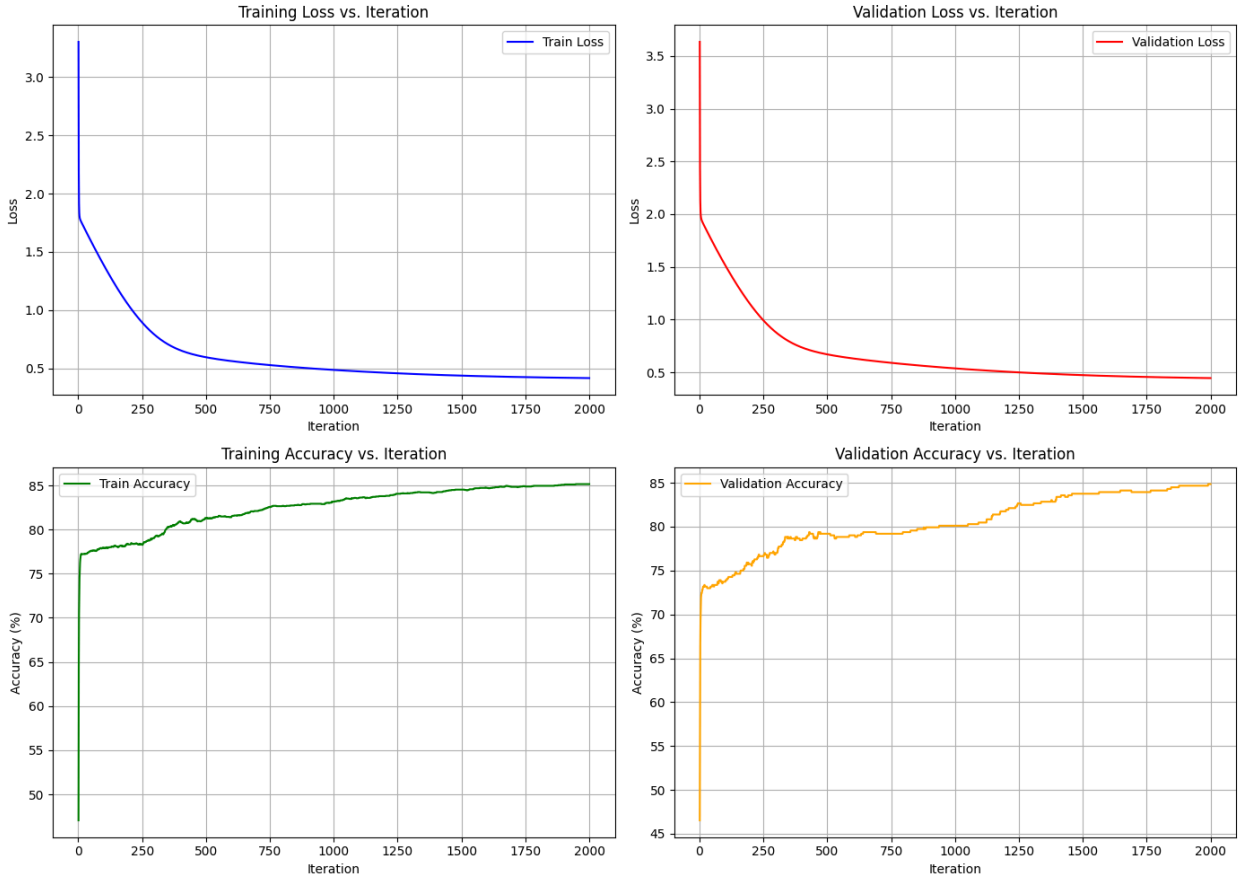
Note : R(f1) = 1391.33

2. (15 points) Section B (Scratch Implementation)

Implement Logistic Regression in the given dataset. You need to implement Gradient Descent from scratch, meaning you cannot use any libraries for training the model (You may use libraries like NumPy for other purposes, but not for training the model). Split the dataset into 70:15:15 (train: test: validation). The loss function to be used is Cross entropy loss.

Dataset: Heart Disease

(a) (3 marks) Implement Logistic Regression using Batch Gradient Descent. Plot training loss vs. iteration, validation loss vs. iteration, training accuracy vs. iteration, and validation accuracy vs. iteration. Comment on the convergence of the model. Compare and analyze the plots.



At convergence the metrics are:
Train Loss: 0.4377, Train Accuracy: 84.53%, Val Loss: 0.4726, Val Accuracy: 83.76%

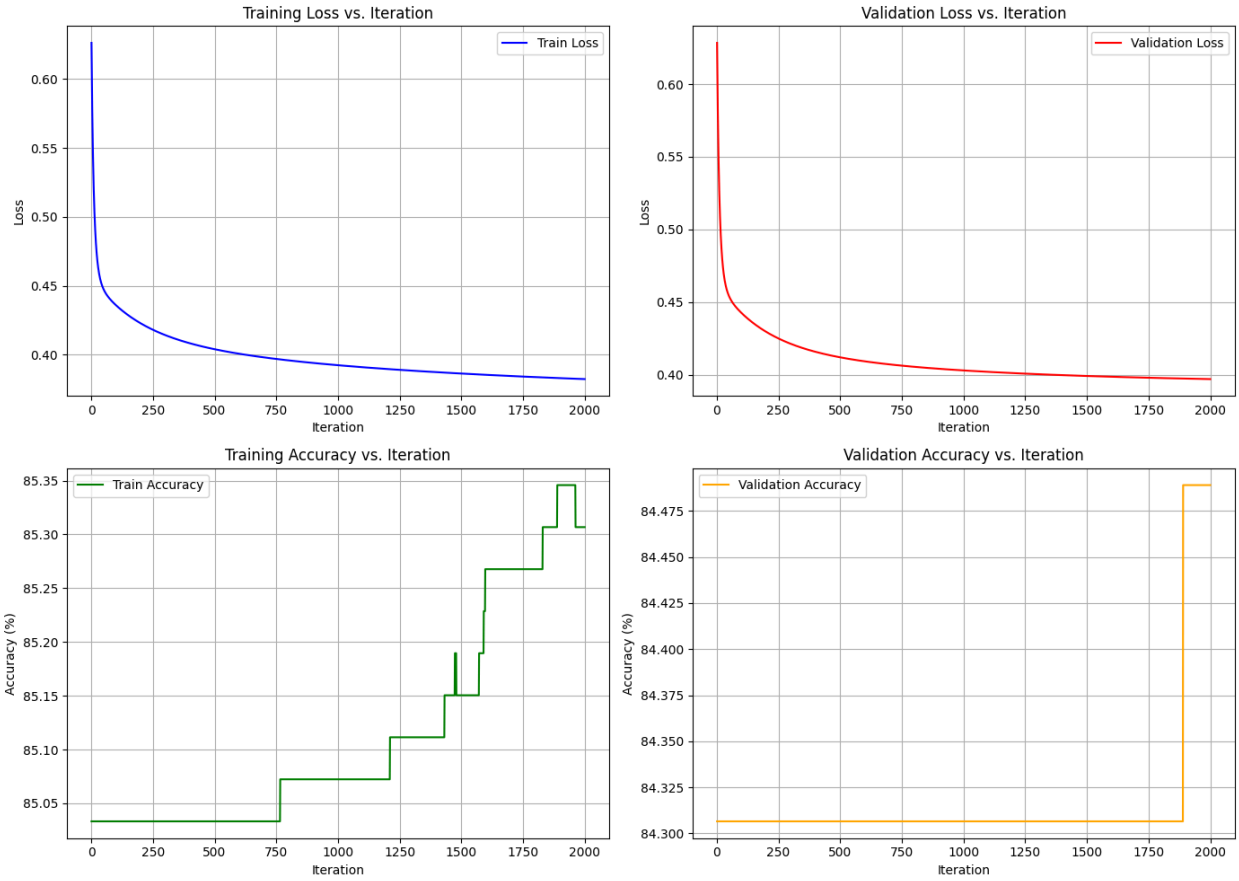
Train loss and validation loss both decrease smoothly and steadily indicating good generalization and minimal overfitting.

Train accuracy is slightly higher than validation accuracy, which is expected. However, the difference is very small, indicating the model is not overfitting.

The model converges well and there are no significant signs of overfitting.

(b) (2 marks) Investigate and compare the performance of the model with

different feature scaling methods: Min-max scaling and No scaling. Plot the loss vs. iteration for each method and discuss the impact of feature scaling on model convergence.



If the learning rate is small, it leads to “straight line” like graphs, because the model doesn’t learn across the epochs.

If the learning rate is increased (to 0.01), it leads to the graphs above.

(c) (2 marks) Calculate and present the confusion matrix for the validation set. Report precision, recall, F1 score, and ROC-AUC score for the model based on the validation set. Comment on how these metrics provide insight into the model’s performance.

Confusion Matrix:
[[455 7]
[76 10]]

Precision: 0.5882 - measure of accuracy of positive predictions

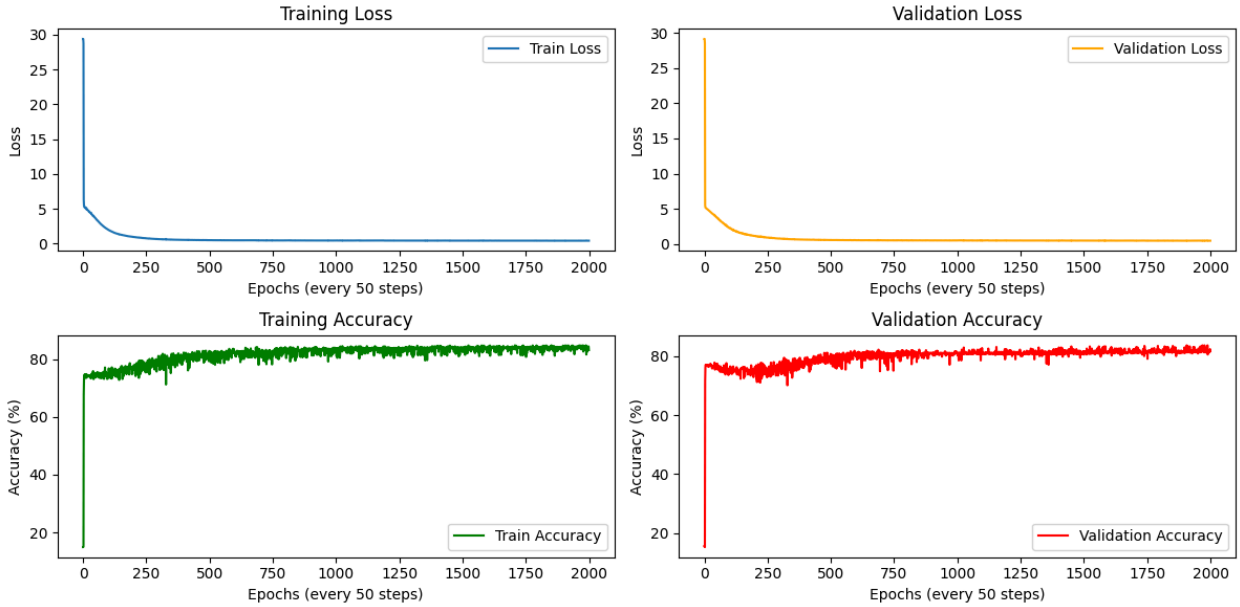
Recall: 0.1163 - measure of completeness of positive predictions

F1 Score: 0.1942 - the harmonic mean of the precision and recall - measure of reliability

ROC-AUC Score: 0.5592 - measure of how model produces relative scores to discriminate between positive or negative instances across all classification thresholds

(d) (3 marks) Implement and compare the following optimisation algorithms: Stochastic Gradient Descent and Mini-Batch Gradient Descent (with varying batch sizes, at least 2). Plot and compare the loss vs. iteration and accuracy vs. iteration for each method. Discuss the trade-offs in terms of convergence speed and stability between these methods.

Stochastic Gradient Descent:



Train Loss: 0.4283, Train Accuracy: 83.12%, Val Loss: 0.4780, Val Accuracy: 81.57%

Confusion Matrix:

[[435 27]

[74 12]]

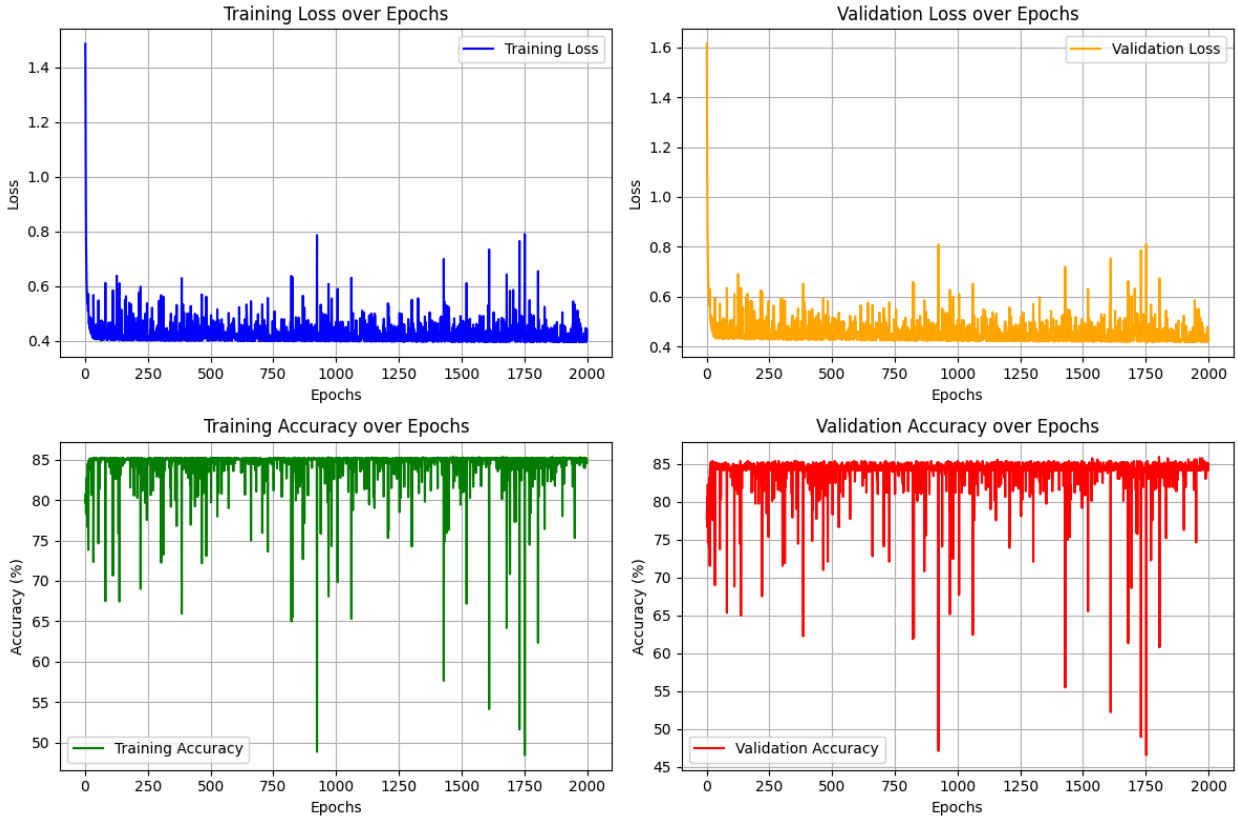
Precision: 0.3077

Recall: 0.1395

F1 Score: 0.1920

ROC-AUC Score: 0.5858

Mini Batch Gradient Descent: (Batch size = 32)



Confusion Matrix:

[[452 10]

[72 14]]

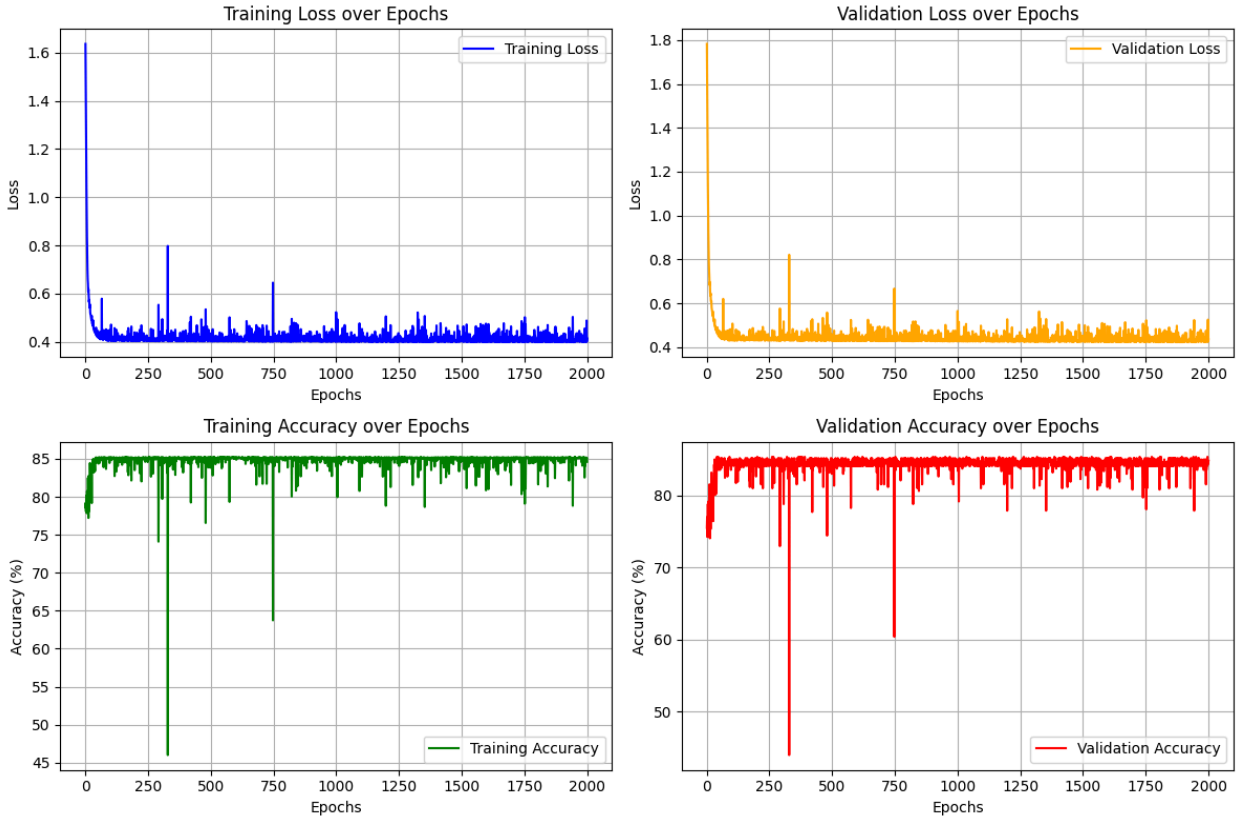
Precision: 0.5833

Recall: 0.1628

F1 Score: 0.2545

ROC-AUC Score: 0.6341

Mini Batch Gradient Descent (Batch Size = 64)



Confusion Matrix:

[[450 12]

[71 15]]

Precision: 0.5556

Recall: 0.1744

F1 Score: 0.2655

ROC-AUC Score: 0.6241

TRADE - OFFS :

SGD converges faster initially as it updates the weights after each data point, resulting in a rapid decrease in loss. However, it can be less stable due to high variance in updates, though in this case, the loss and accuracy curves are smooth, indicating stable convergence.

MBGD converges more slowly since updates occur after processing a batch, not a single point. The loss and accuracy curves in MBGD show significant fluctuations, reflecting instability from noisy updates.

MBGD typically balances the high variance of SGD and the slower, more stable convergence of full-batch gradient descent, but it still experiences notable noise.

The choice depends on what is important - convergence (SGD) or less variance(MBGD).

(e) (2 marks) Implement k-fold cross-validation (with k=5) to assess the robustness of your model. Report the average and standard deviation for

accuracy, precision, recall, and F1 score across the folds. Discuss the stability and variance of the model's performance across different folds.

Average Accuracy ± Std Dev: 84.85% ± 1.16

Average Precision ± Std Dev: 0.61 ± 0.27

Average Recall ± Std Dev: 0.02 ± 0.01

Average F1 Score ± Std Dev: 0.04 ± 0.02

Across different folds, accuracy starts around ~45% and increases over time. Both training and validation accuracy fluctuate and have more significant variance across folds. This suggests that the model at first undergoes more significant changes in the weights, leading to higher variance before stabilization.

After ~5000 epochs, training accuracy stabilizes consistently (85.23% in Fold 1, 84.44% in Fold 2, 85.06% in Fold 3,etc.), indicating the model reaches stable state, with convergence and low variance in training performance across folds.

The validation accuracy stabilizes consistently (Fold 1:83.61% Fold 2: 86.73%. Fold 3: 84.40% etc.) indicating the model generalizes well, with consistent validation performance across folds.

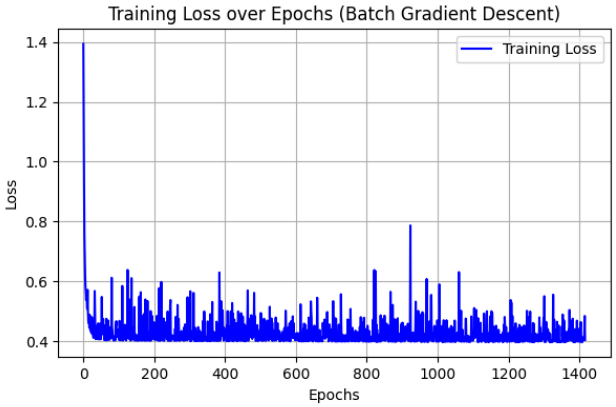
Training accuracy stabilizes earlier (~5000 epochs), while validation accuracy takes slightly longer, suggesting the model fits well to the training data but adjustments continue on the validation data until later epochs.

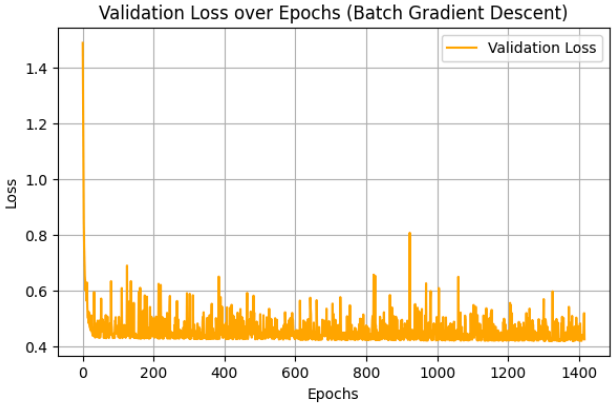
(f) (3 marks) Implement early stopping in your best Gradient Descent method to avoid overfitting. Define and use appropriate stopping criteria. Experiment with different learning rates and regularization techniques (L1 and L2). Plot and compare the performance with and without early stopping. Analyze the effect of early stopping on overfitting and generalization.

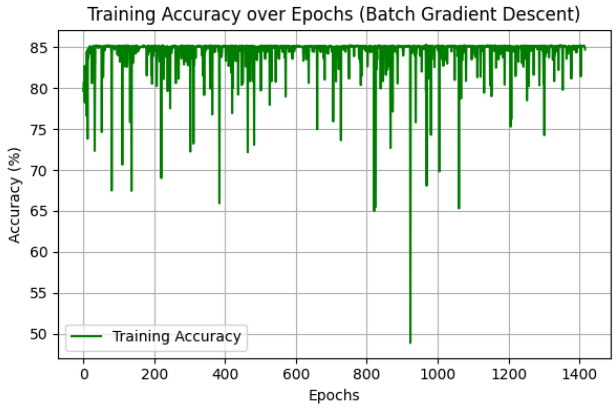
The graph for without early stopping caan be found at 2 (d)

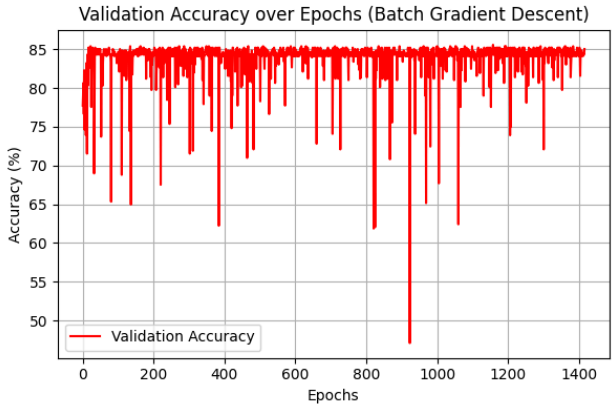
Mini Batch had the best F1 Score of 0.2

Learning rate = 0.0001, L1 = 0, L2 =0.1

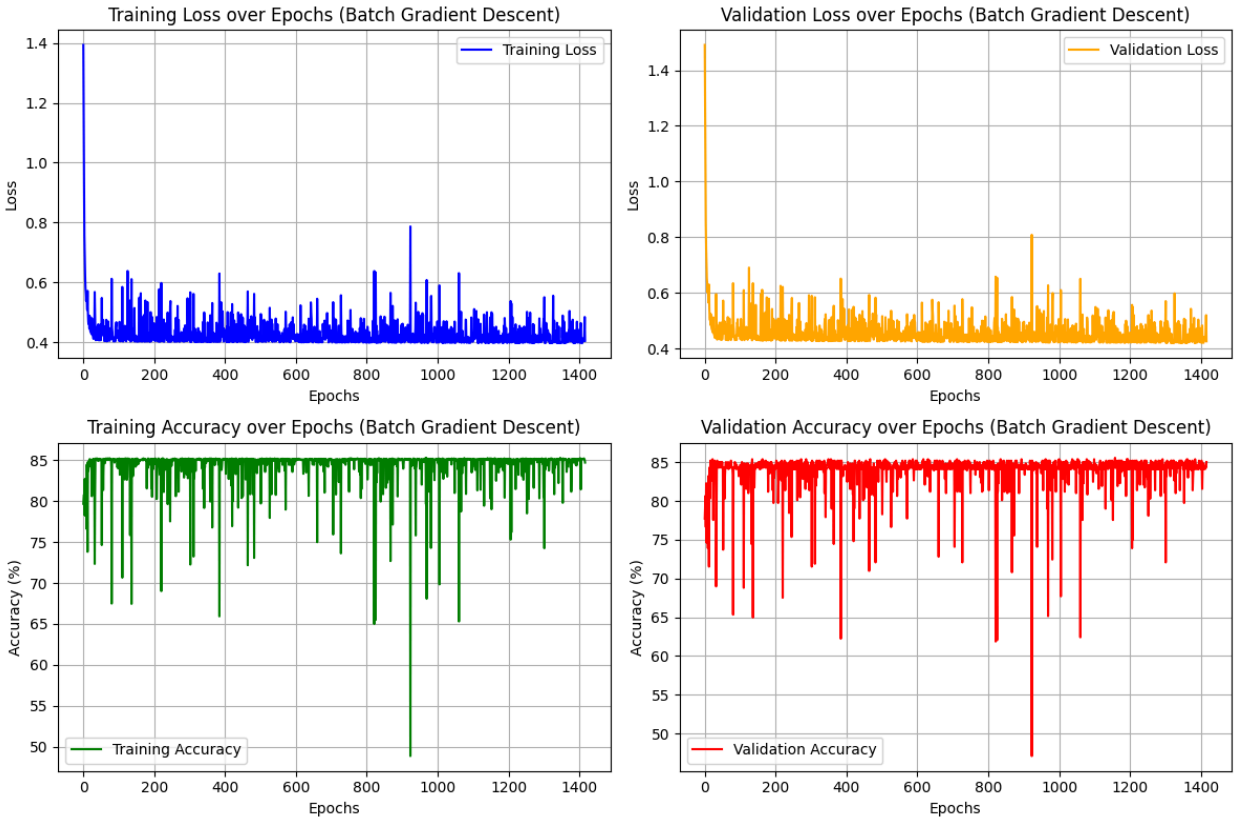




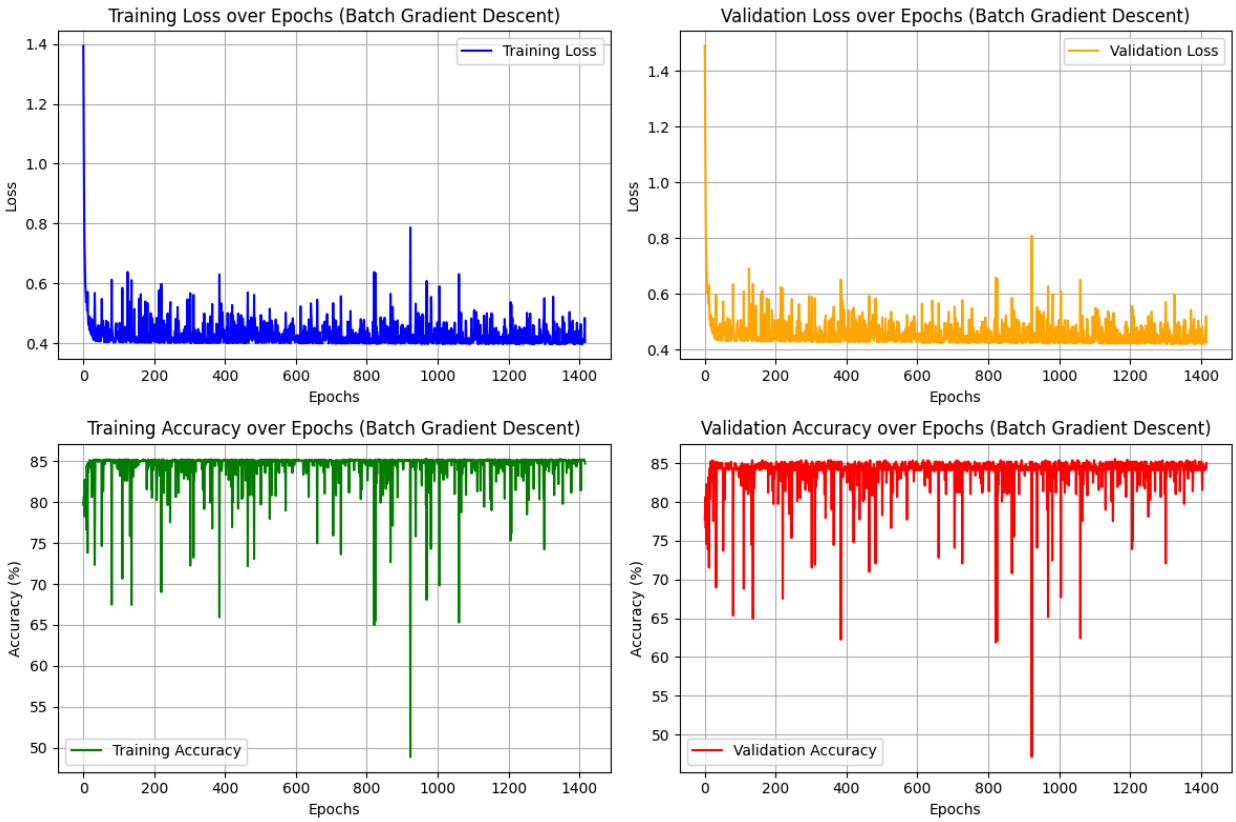




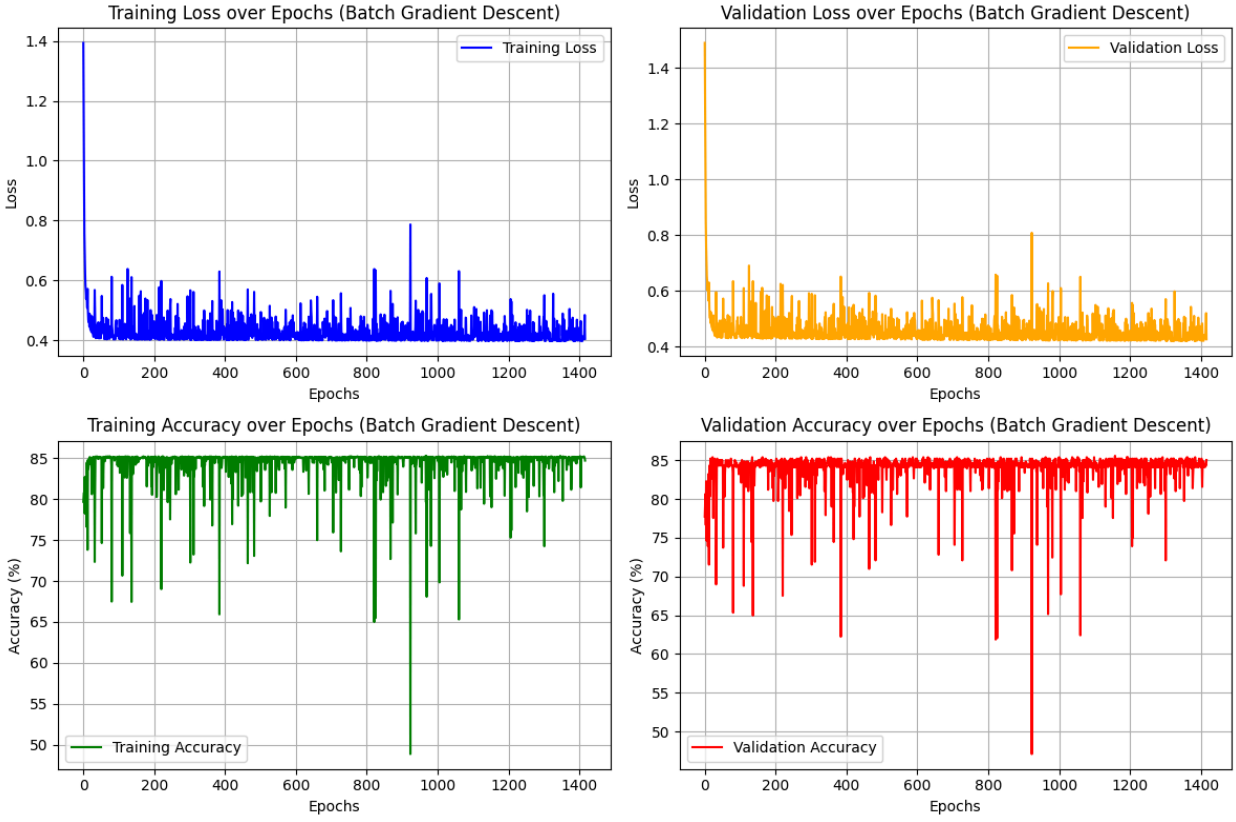
Learning Rate = 0.0001, L1 = 0.1, L2=0



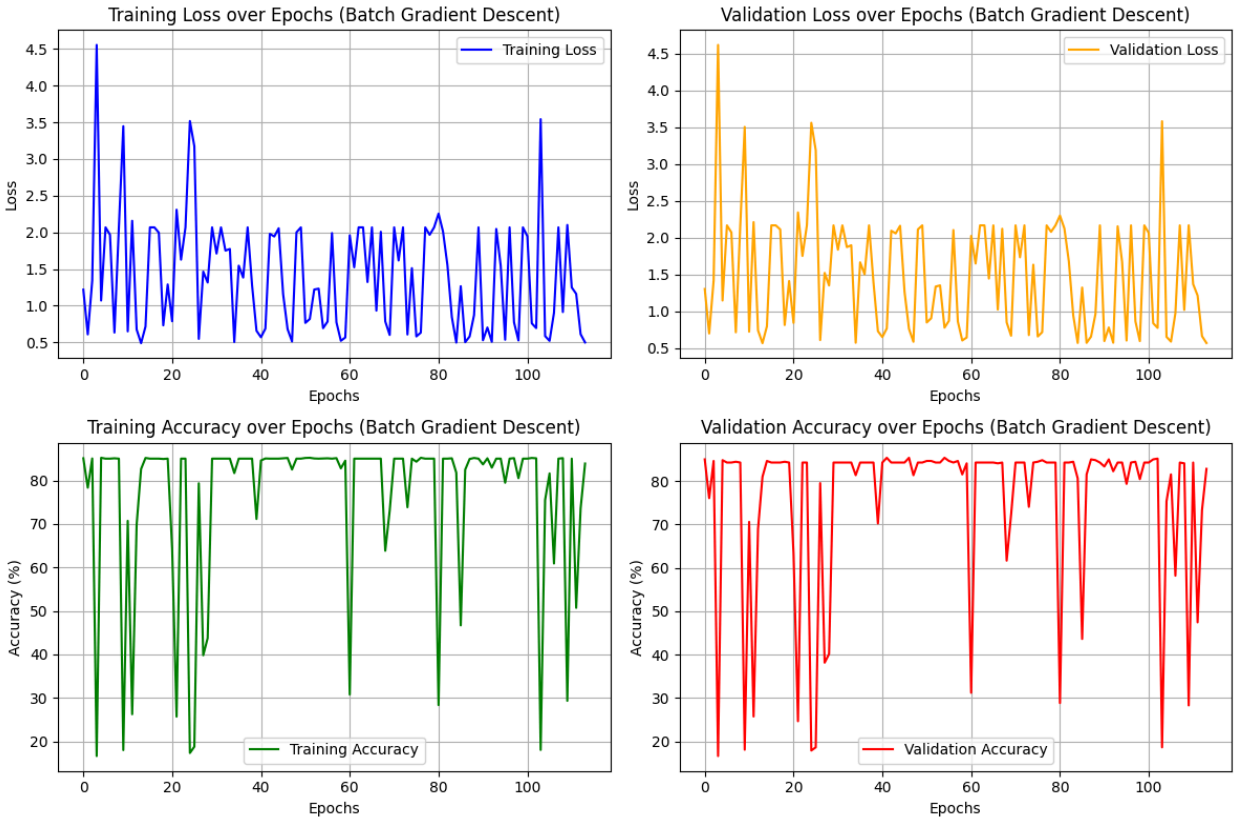
Learning rate = 0.0001, L1 = 0.01, L2 = 0



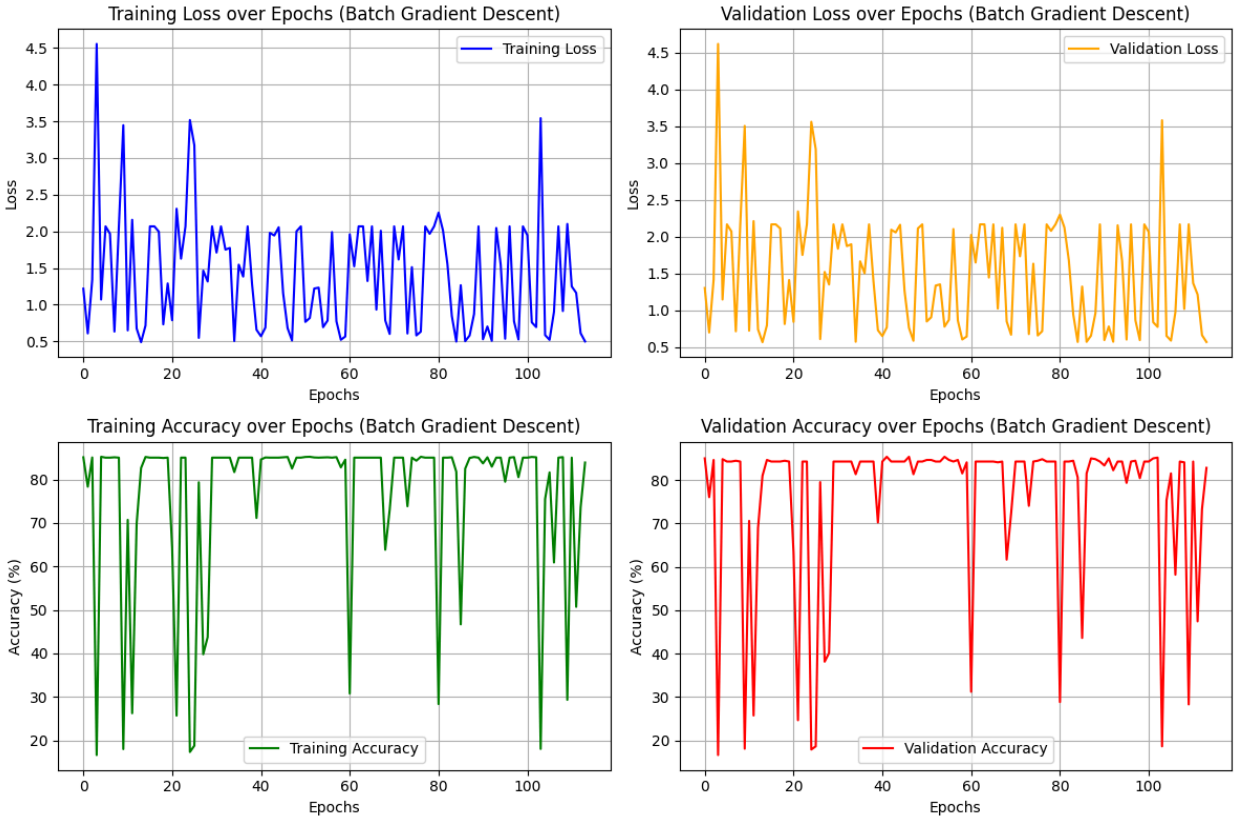
Learning rate = 0.0001, L1 =0, L2 = 0.01



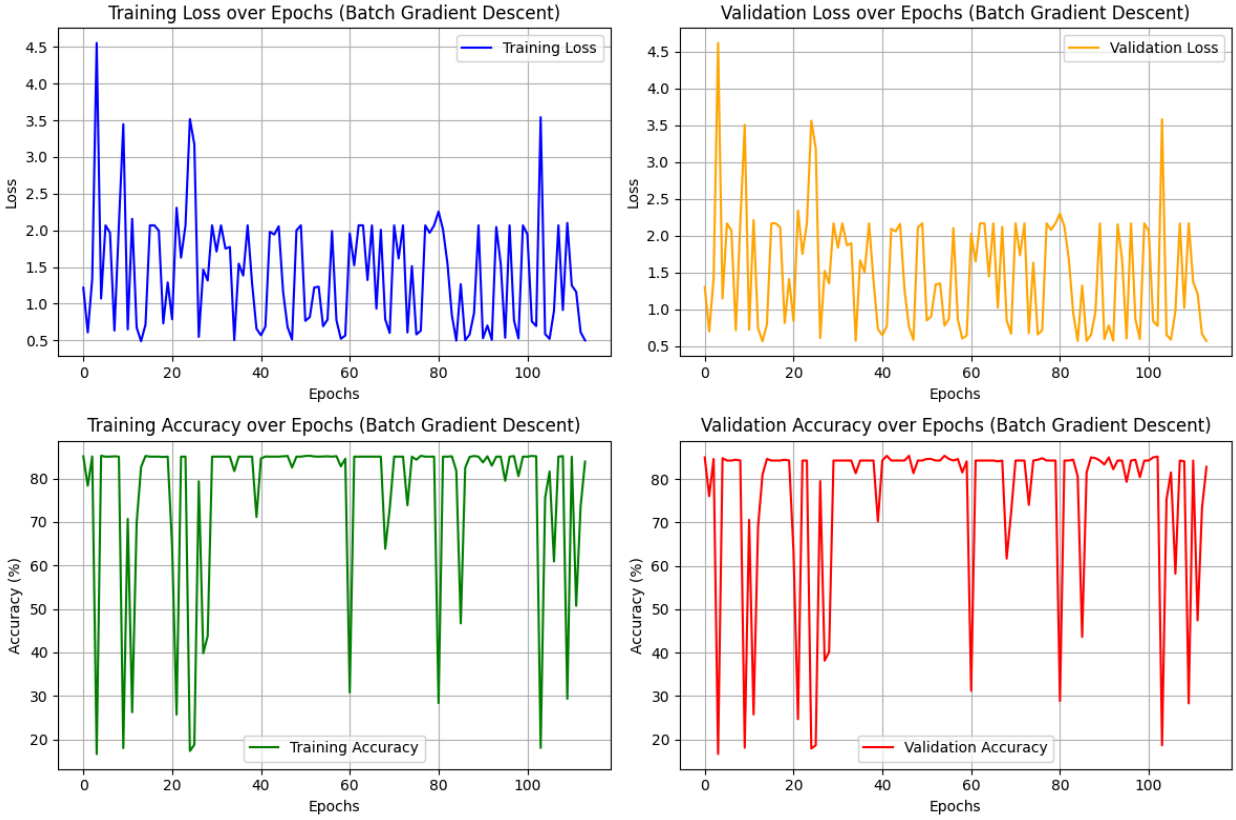
Learning rate = 0.0005, L1 = 0.1, L2 =0



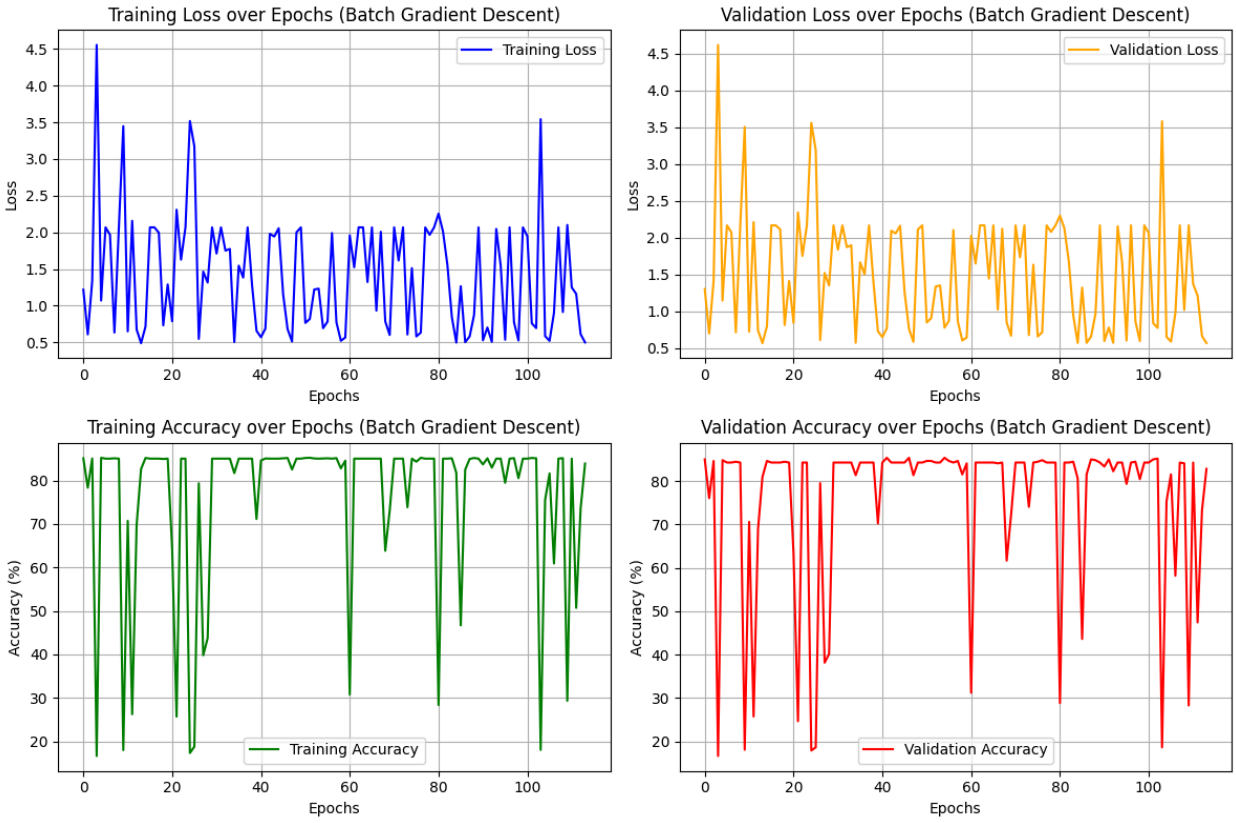
Learning rate = 0.0005, L1 = 0, L2 =0.1



Learning Rate = 0.0005, L1 = 0.01, L2 = 0



Learning rate = 0.0005, L1 = 0, L2 = 0.01



The graphs indicate:

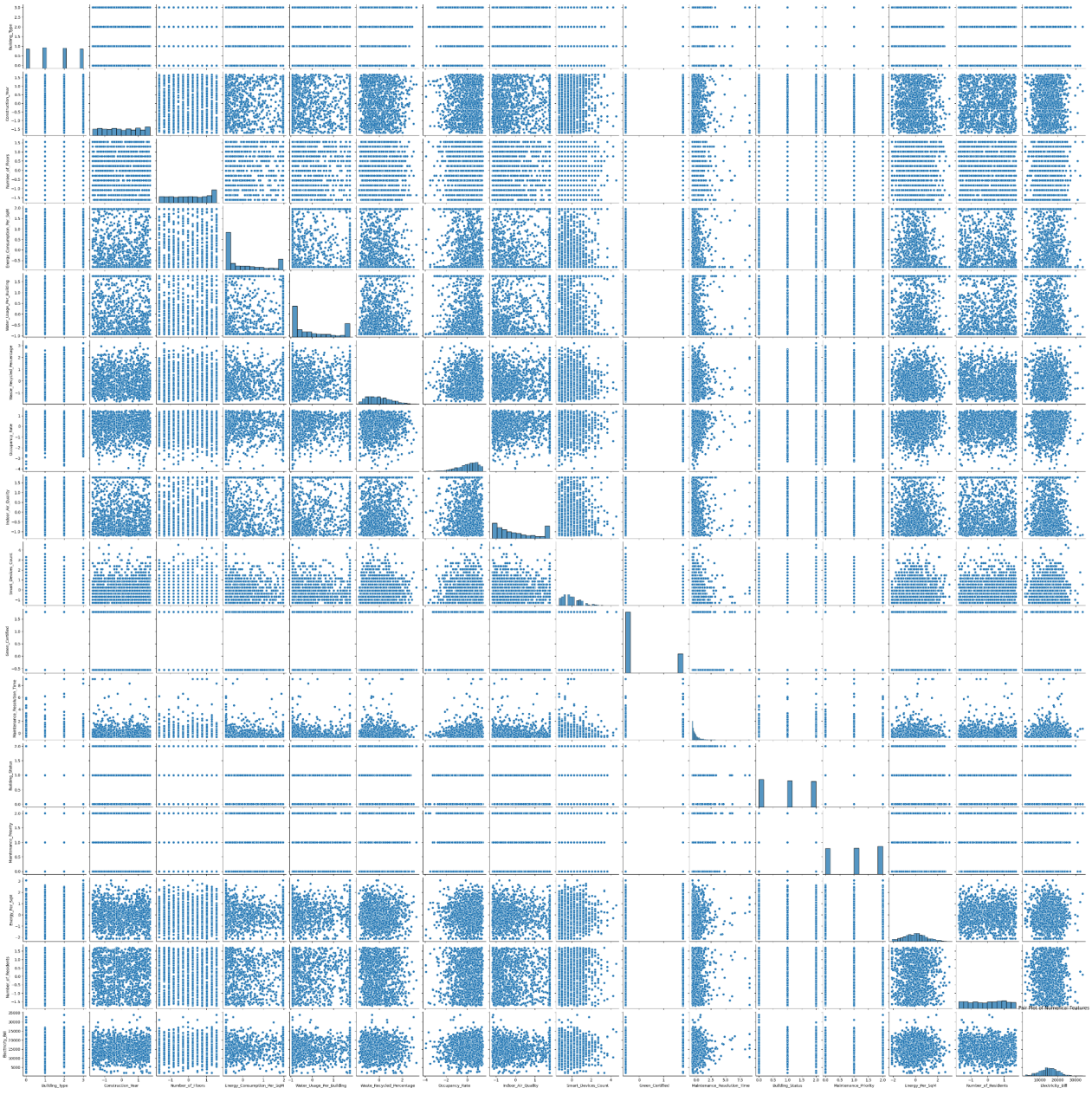
Increasing the learning rate causes instability/noise in the graph

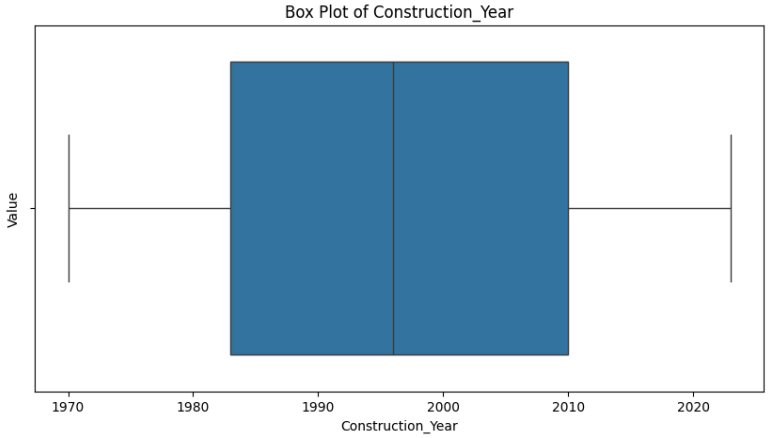
L1 and L2 have similar effects, but sometimes L1 converges earlier.

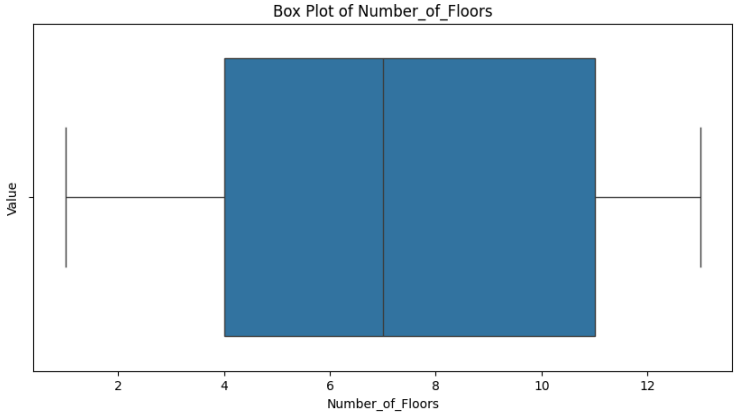
3. (15 points) Section C (Algorithm implementation using packages)

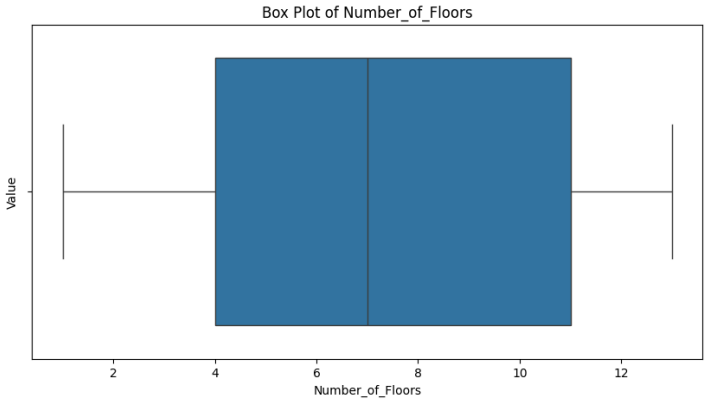
Split the given dataset into 80:20 (train: test) and perform the following tasks: Dataset: Electricity Bill Dataset

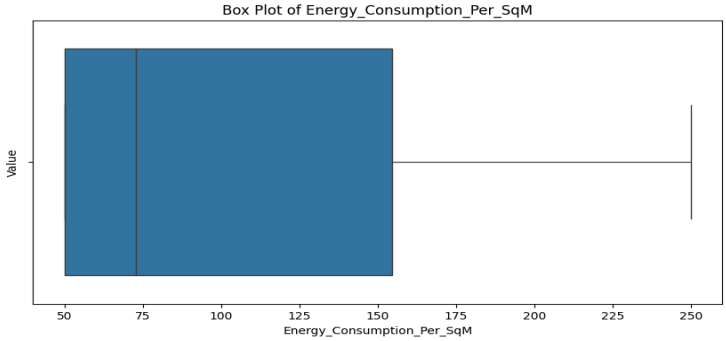
(a) (2.5 marks) Perform EDA by creating pair plots, box plots, violin plots, count plots for categorical features, and a correlation heatmap. Based on these visualizations, provide at least five insights on the dataset.

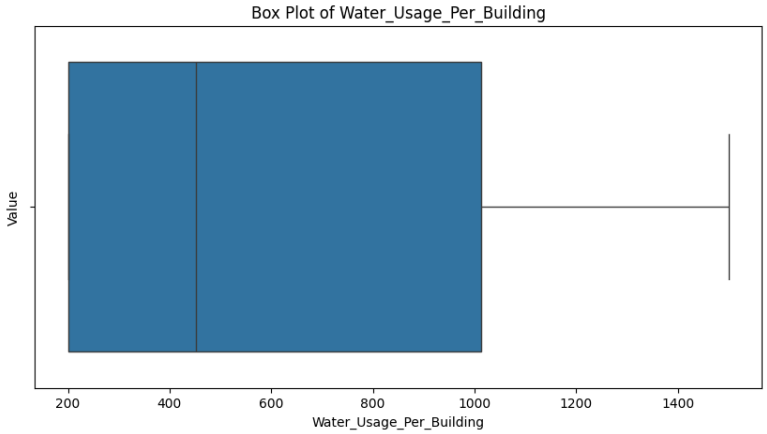

Pair Plot of Numerical Features
Building_Type
Construction_Year
Number_of_Floors
Energy_Consumption_Per_SqM
Water_Usage_Per_Building
Waste_Recycled_Percentage
Occupancy_Rate
Indoor_Air_Quality
Smart_Devices_Count
Green_Certified
Maintenance_Resolution_Time
Building_Status
Maintenance_Priority
Energy_Per_SqM
Number_of_Residents
Electricity_Bill

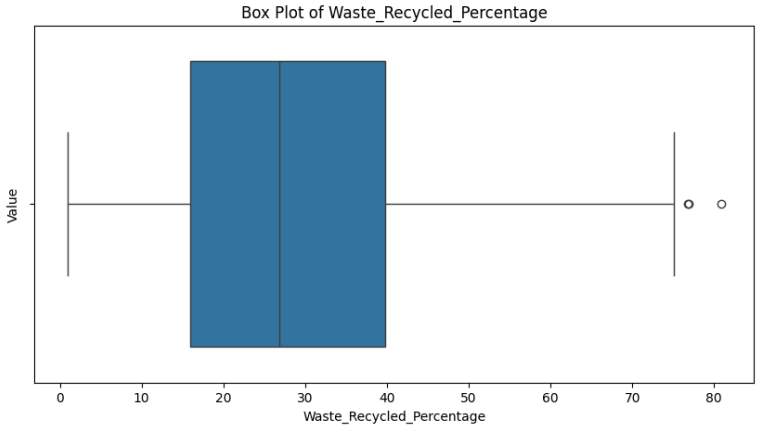

Box Plot of Construction_Year
Value
1970
1980
1990
2000
2010
2020
Construction_Year

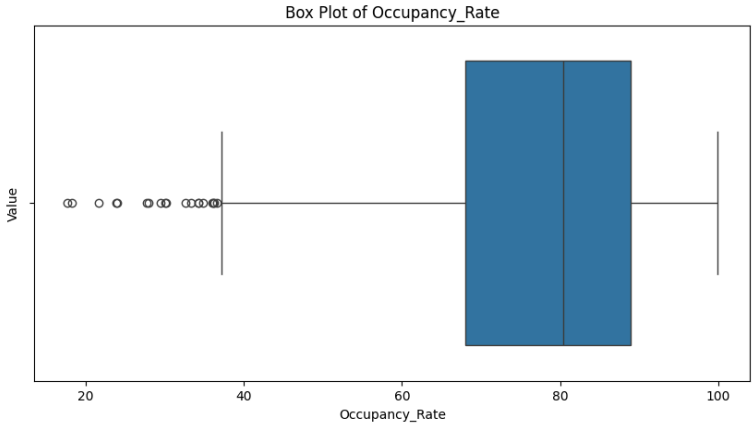

Box Plot of Number_of_Floors
Value
2
4
6
8
10
12
Number_of_Floors

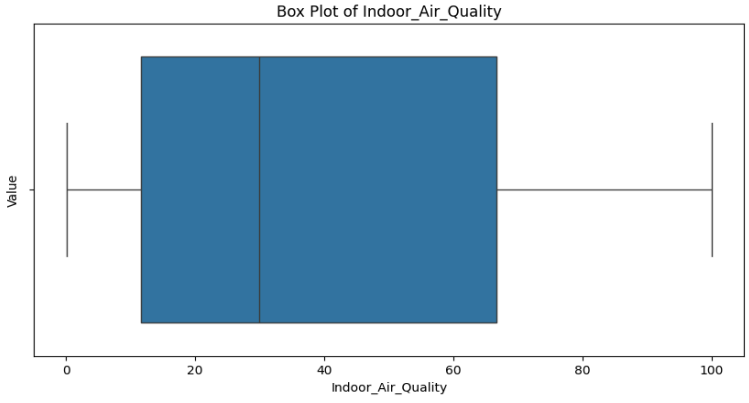

Box Plot of Number_of_Floors
Value
2
4
6
8
10
12
Number_of_Floors

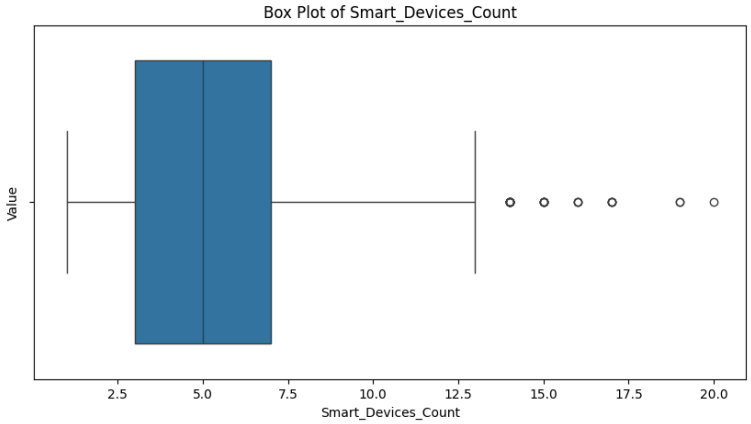

Box Plot of Energy_Consumption_Per_SqM
Value
50
75
100
125
150
175
200
225
250
Energy_Consumption_Per_SqM

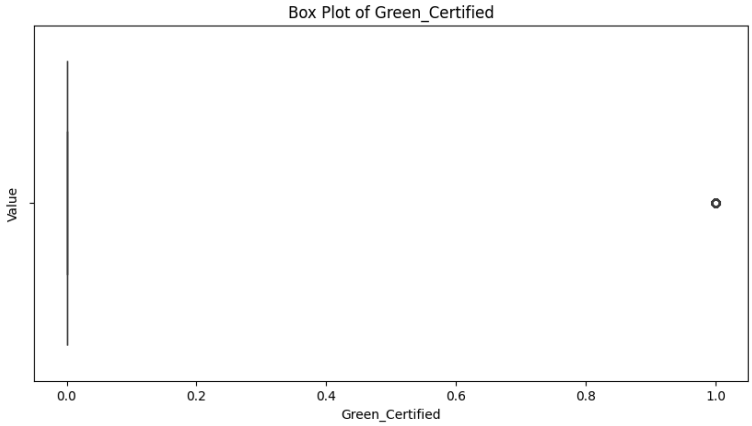

Box Plot of Water_Usage_Per_Building
Value
200
400
600
800
1000
1200
1400
Water_Usage_Per_Building

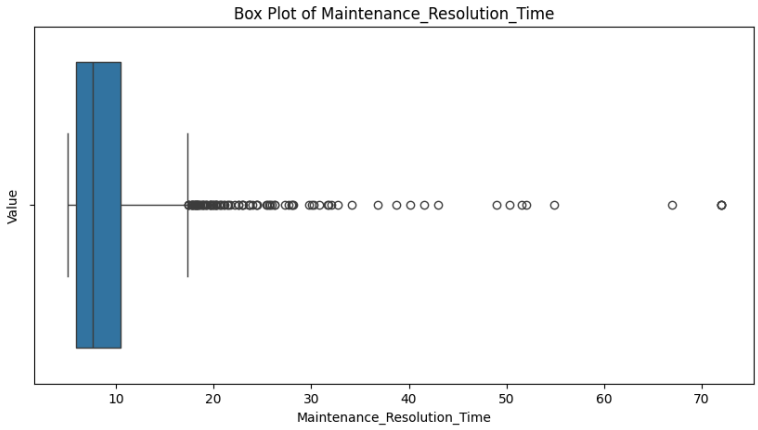

Box Plot of Waste_Recycled_Percentage
Value
0
10
20
30
40
50
60
70
80
Waste_Recycled_Percentage

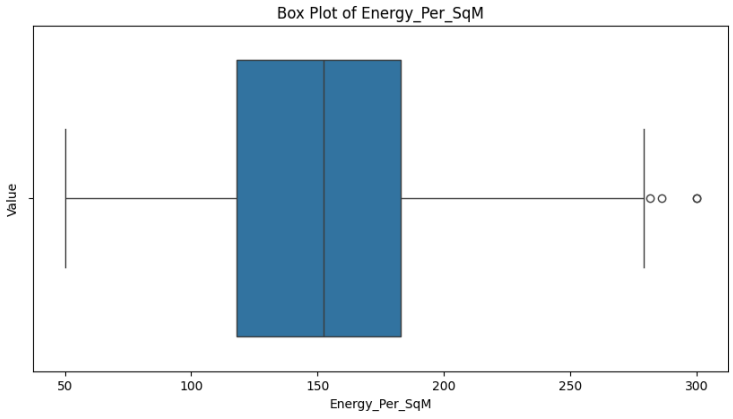

Box Plot of Occupancy_Rate
Value
20
40
60
80
100
Occupancy_Rate

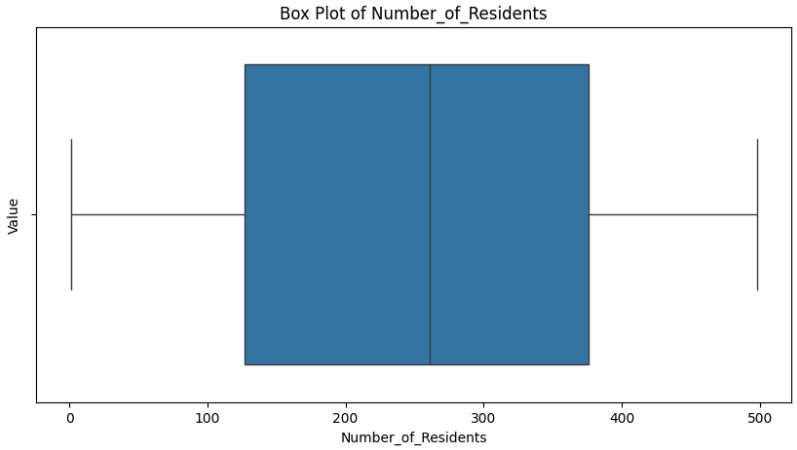

Box Plot of Indoor_Air_Quality
Value
0
20
40
60
80
100
Indoor_Air_Quality

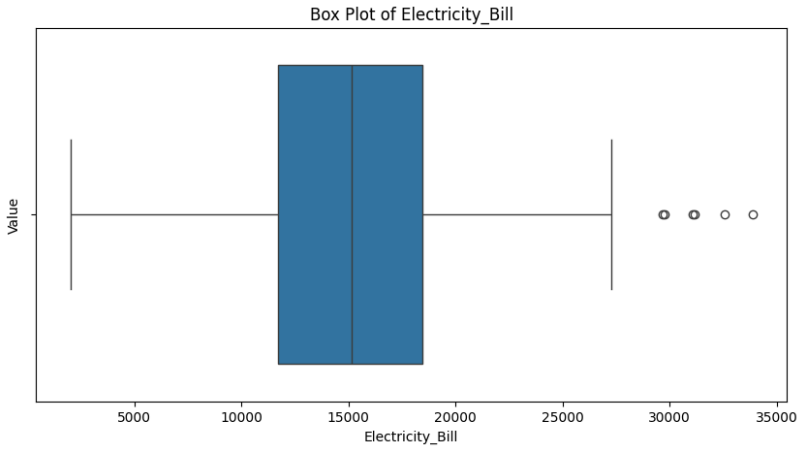

Box Plot of Smart_Devices_Count
Value
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Smart_Devices_Count

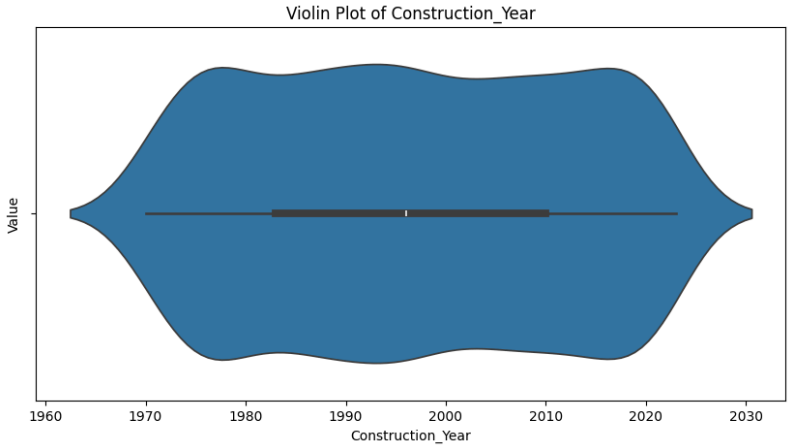

Box Plot of Green_Certified
Value
0.0
0.2
0.4
0.6
0.8
1.0
Green_Certified

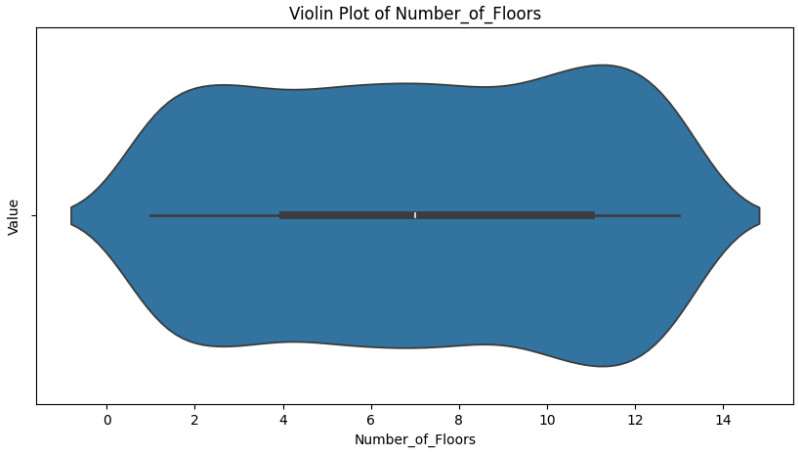

Box Plot of Maintenance_Resolution_Time
Value
10
20
30
40
50
60
70
Maintenance_Resolution_Time

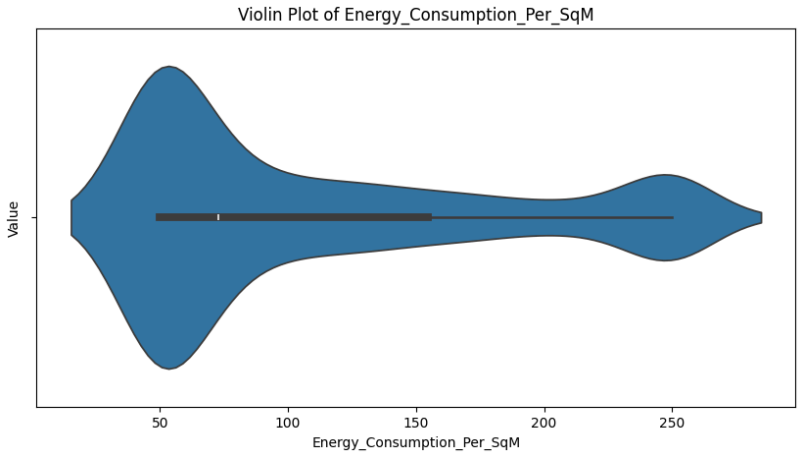

Box Plot of Energy_Per_SqM
Value
50
100
150
200
250
300
Energy_Per_SqM

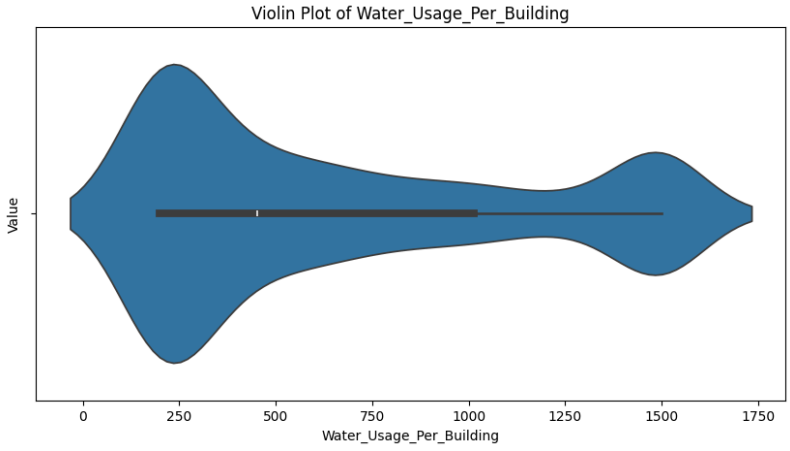

Box Plot of Number_of_Residents
Value
0
100
200
300
400
500
Number_of_Residents

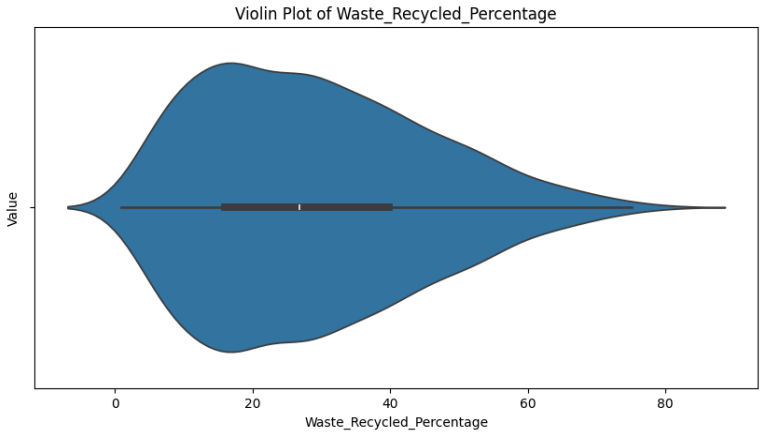

Box Plot of Electricity_Bill
Value
5000
10000
15000
20000
25000
30000
35000
Electricity_Bill

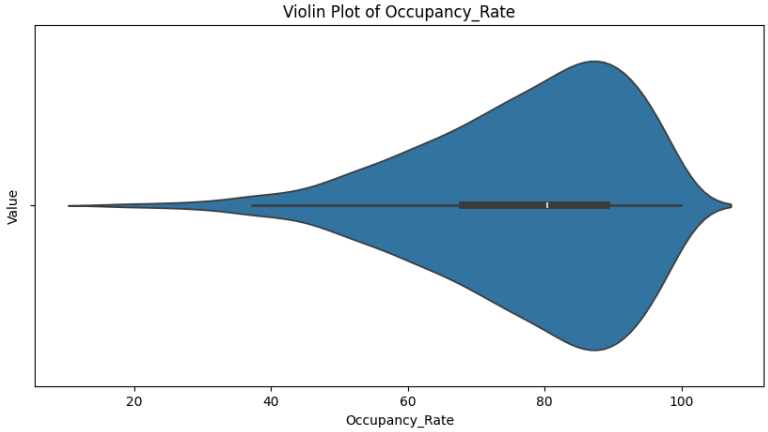

Violin Plot of Construction_Year
Value
1960
1970
1980
1990
2000
2010
2020
2030
Construction_Year

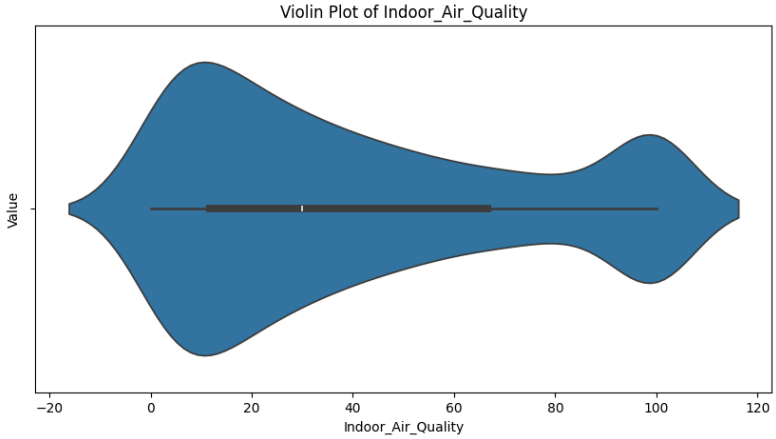

Violin Plot of Number_of_Floors
Value
0
2
4
6
8
10
12
14
Number_of_Floors

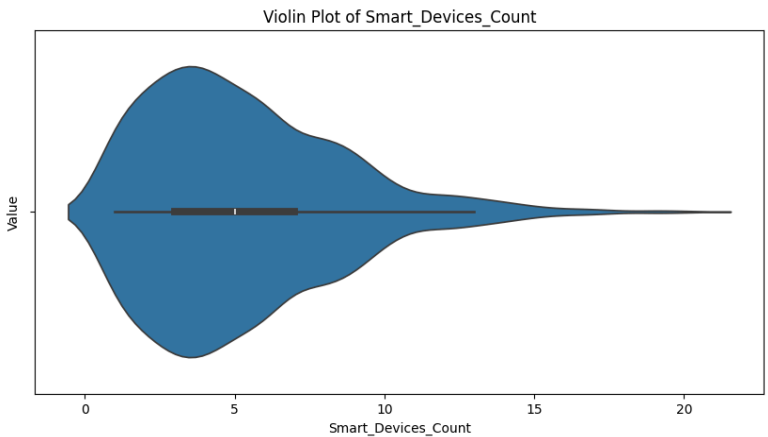

Violin Plot of Energy_Consumption_Per_SqM
Value
50
100
150
200
250
Energy_Consumption_Per_SqM

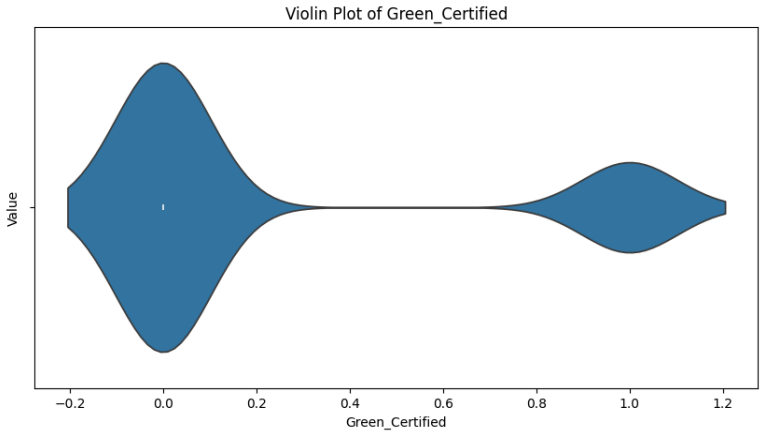

Violin Plot of Water_Usage_Per_Building
Value
0
250
500
750
1000
1250
1500
1750
Water_Usage_Per_Building

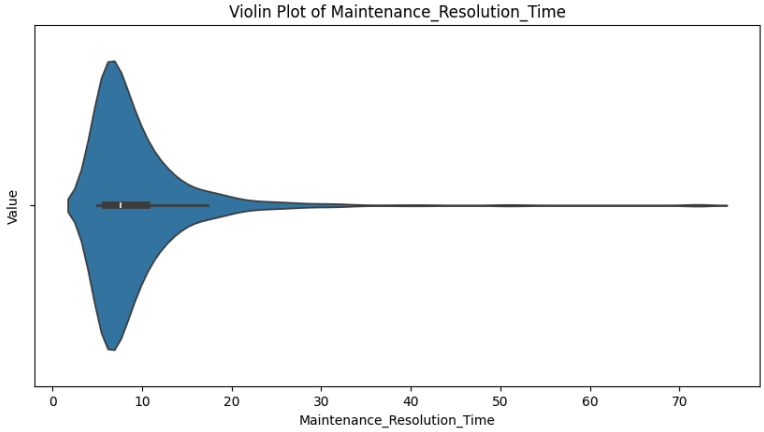

Violin Plot of Waste_Recycled_Percentage
Value
0
20
40
60
80
Waste_Recycled_Percentage

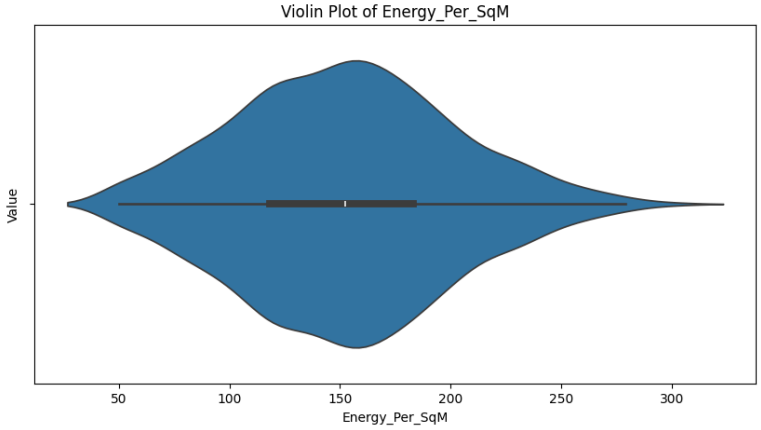

Violin Plot of Occupancy_Rate
Value
20
40
60
80
100
Occupancy_Rate

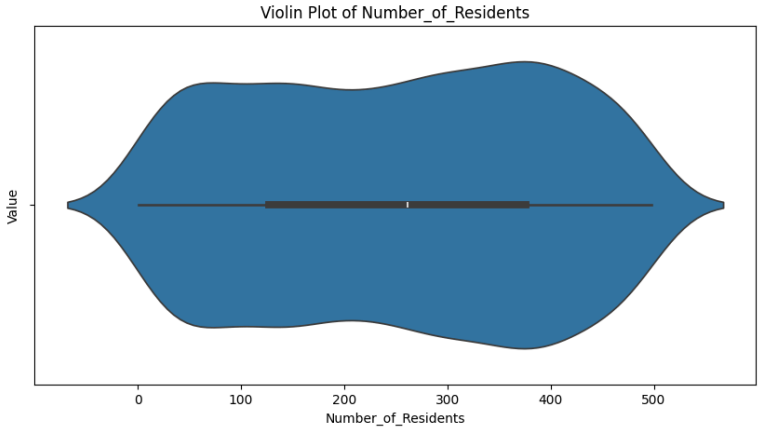

Violin Plot of Indoor_Air_Quality
Value
−20
0
20
40
60
80
100
120
Indoor_Air_Quality

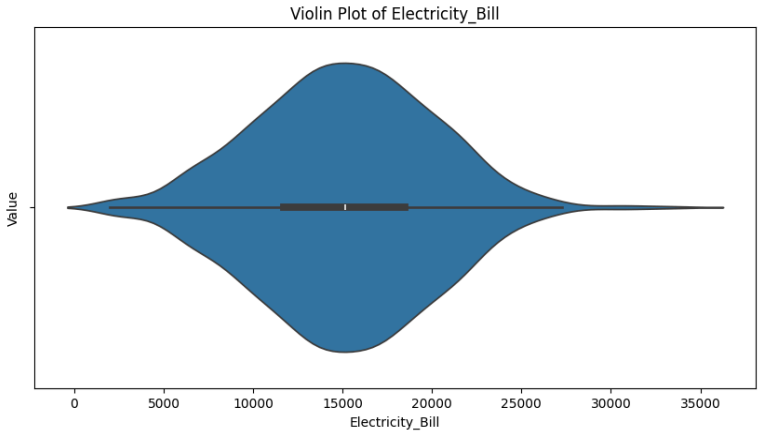

Violin Plot of Smart_Devices_Count
Value
0
5
10
15
20
Smart_Devices_Count

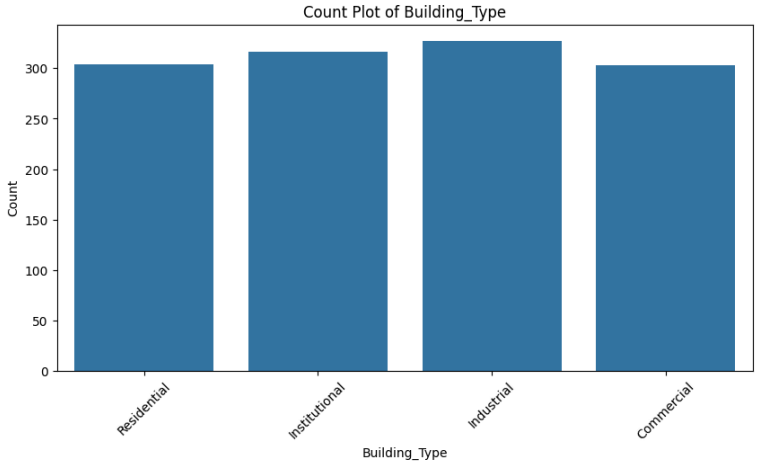

Violin Plot of Green_Certified
Value
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Green_Certified

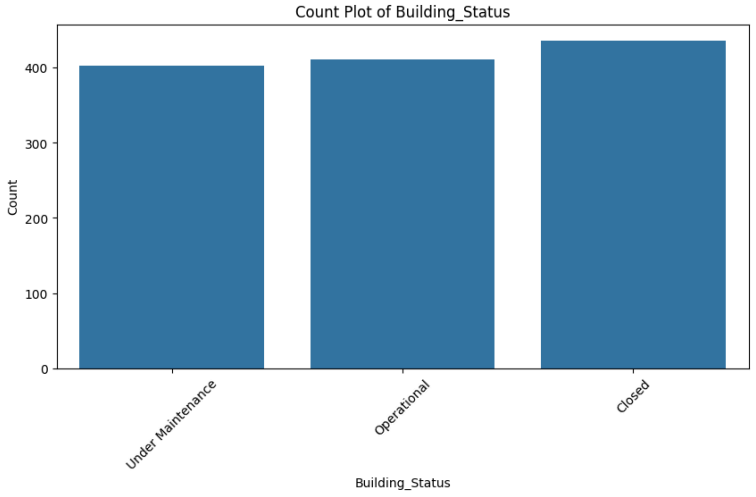

Violin Plot of Maintenance_Resolution_Time
Value
0
10
20
30
40
50
60
70
Maintenance_Resolution_Time

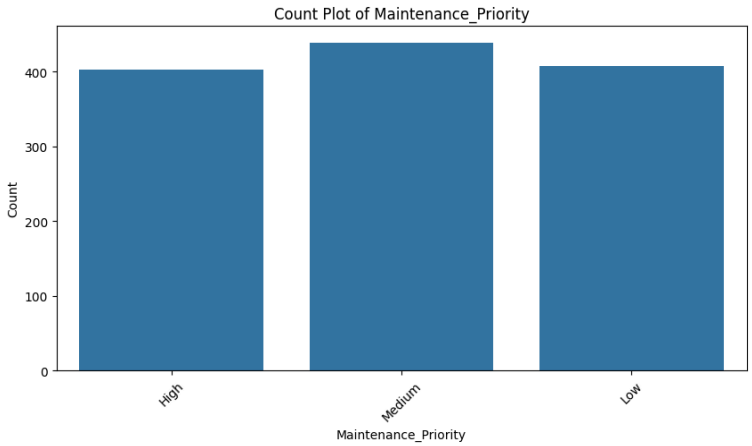

Violin Plot of Energy_Per_SqM
Value
50
100
150
200
250
300
Energy_Per_SqM

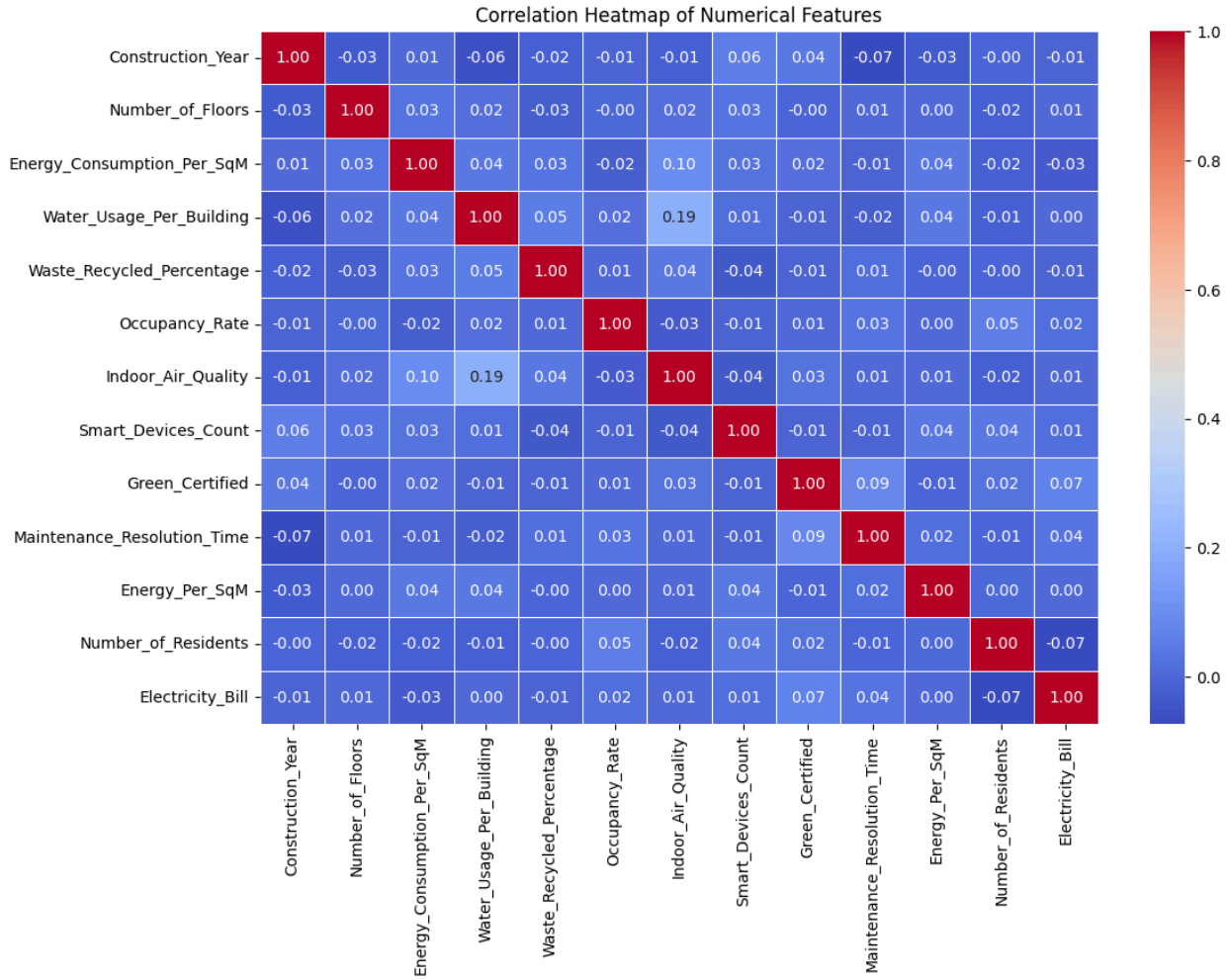

Violin Plot of Number_of_Residents
Value
0
100
200
300
400
500
Number_of_Residents


Violin Plot of Electricity_Bill
Value
0
5000
10000
15000
20000
25000
30000
35000
Electricity_Bill


Count Plot of Building_Type
Count
0
50
100
150
200
250
300
Residential
Institutional
Industrial
Commercial
Building_Type


Count Plot of Building_Status
Count
0
100
200
300
400
Under Maintenance
Operational
Closed
Building_Status





Inferences:

Based on the pair plots and heatmap correlation, if the number of residents are more, the energy consumption per sq m is less

There is a positive correlation between Indoor Air Quality and Green Certified

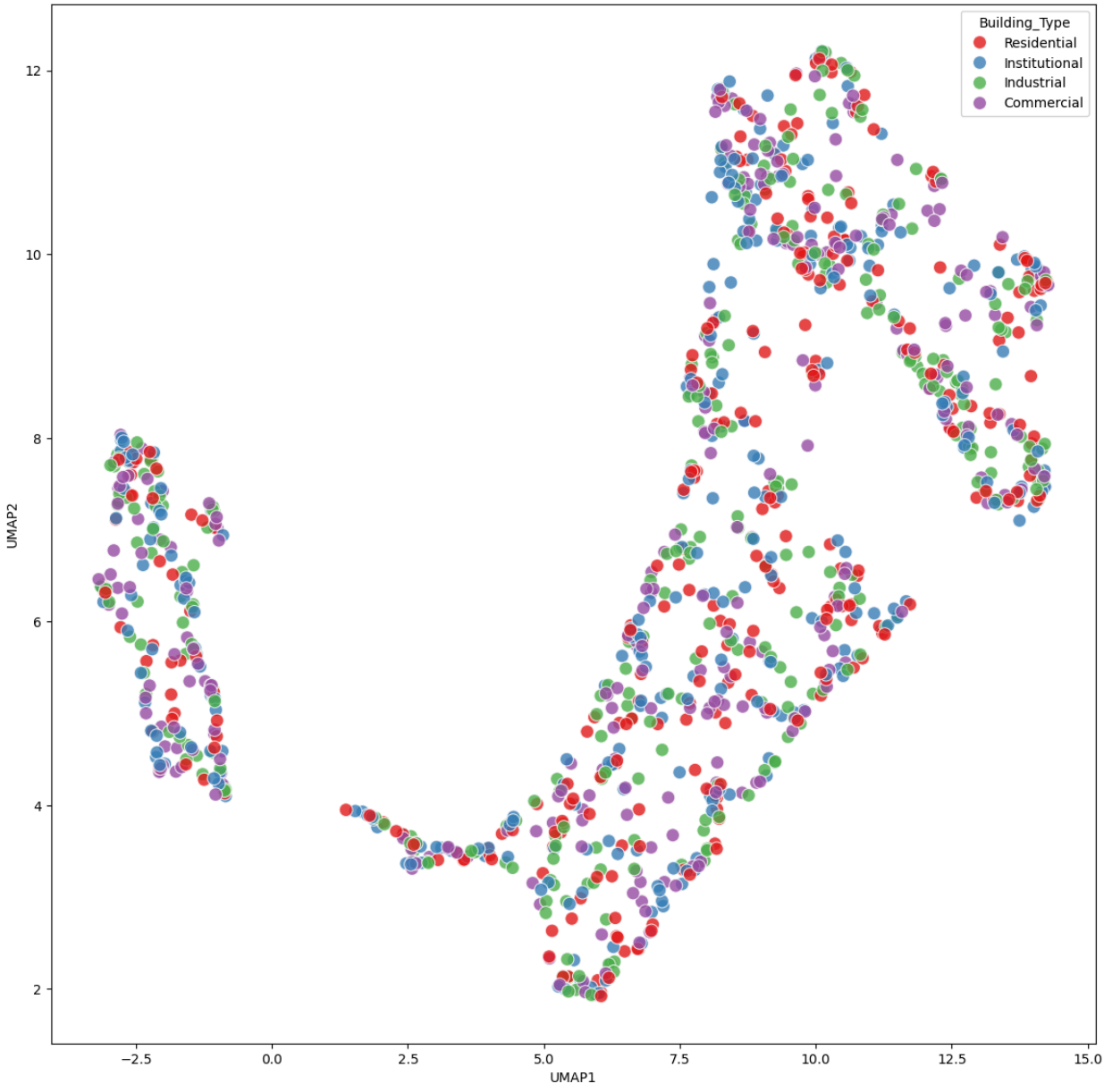
There is a positive correlation between Number of Floors and Indoor Air Quality

There is a negative correlation between Construction Year and Number of Floors, meaning

the older homes have less floors.

There is a positive correlation between Energy Consumption per sq m and Water Usage

(b) (1 marks)Use the Uniform Manifold Approximation and Projection (UMAP) algorithm to reduce the data dimensions to 2 and plot the resulting data as a scatter plot. Comment on the separability and clustering of the data after dimensionality reduction.



There are two clusters and they are separable.

(c) (2.5 marks) Perform the necessary pre-processing steps, including handling missing values and normalizing numerical features. For categorical features, use LabelEncoding. Apply Linear Regression on the preprocessed data. Report Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R2 score, Adjusted R2 score, and Mean Absolute Error (MAE) on the train and test data.

Training set evaluation:

MSE: 24475013.1685

RMSE: 4947.2228

R2 Score: 0.0139

Adjusted R2 Score: -0.0011

MAE: 4006.3285

Test set evaluation:

MSE: 24278016.1557

RMSE: 4927.2727

R2 Score: 0.0000

Adjusted R2 Score: -0.0641

MAE: 3842.4093

(d) (2 marks) Perform Recursive Feature Elimination (RFE) or Correlation analysis on the original dataset to select the 3 most important features. Train the regression model using the selected features. Compare the results (MSE, RMSE, R2 score, Adjusted R2 score, MAE) on the train and test dataset with the results obtained in part (c).

Selected features: ['Building_Type', 'Green_Certified', 'Number_of_Residents']

Training set evaluation:

MSE: 24569032.9069

RMSE: 4956.7159

R2 Score: 0.0101

Adjusted R2 Score: 0.0072

MAE: 4006.4734

Test set evaluation:

MSE: 23941409.0630

RMSE: 4892.9959

R2 Score: 0.0139

Adjusted R2 Score: 0.0019

MAE: 3813.9481

MSE of (d) is higher than (c) by ~94019

RMSE of (d) is higher than © by ~9.4931

The R2 score is more or less the same.

The adjusted R2 score of (d) is higher.

The MAE of (d) is less than (c) by ~180

(e) (2 marks) Encode the categorical features of the original dataset using One-Hot Encoding and perform Ridge Regression on the preprocessed data. Report the evaluation metrics (MSE, RMSE, R2 score, Adjusted R2 score, MAE). Compare the results with those obtained in part (c).

Training set evaluation:
MSE: 24188934.3401
RMSE: 4918.2247
R2 Score: 0.0254
Adjusted R2 Score: 0.0066
MAE: 3976.7356

Test set evaluation:
MSE: 24128288.5043
RMSE: 4912.0554
R2 Score: 0.0062
Adjusted R2 Score: -0.0759
MAE: 3797.5126

The MSE of (e) is less than (c) by ~286,079
The RMSE of (e) is less than (c) by ~28
R2 Score and Adj R2 Score are more or less the same.
MAE is more or less the same.

(f) (2 marks) Perform Independent Component Analysis (ICA) on the one-hot encoded dataset and choose the appropriate number of components (try 4, 5, 6, and 8 components). Compare the results (MSE, RMSE, R2 score, Adjusted R2 score, MAE) on the train and test dataset.

Training set evaluation:

MSE: 24718005.3039

RMSE: 4971.7206

R2 Score: 0.0041

Adjusted R2 Score: 0.0001

MAE: 4016.7368

Test set evaluation:

MSE: 24450425.9303

RMSE: 4944.7372

R2 Score: -0.0071

Adjusted R2 Score: -0.0235

MAE: 3857.9908

The difference in train and test datasets are:

- MSE ~ 267,579
- RMSE ~ 25
- R2 Score ~ same
- Adj R2 Score ~ -0.02
- MAE ~ 150

(g) (1.5 marks) Use ElasticNet regularization (which combines L1 and L2) while training a linear model on the preprocessed dataset from part (c). Compare the evaluation metrics (MSE, RMSE, R2 score, Adjusted R2 score, MAE) on the test dataset for different values of the mixing parameter (alpha).

For alpha = 0.1

Training set evaluation:

MSE: 24475793.2838

RMSE: 4947.3016

R2 Score: 0.0139

Adjusted R2 Score: -0.0011

MAE: 4005.3930

Test set evaluation:

MSE: 24278016.1557

RMSE: 4927.2727

R2 Score: 0.0000

Adjusted R2 Score: -0.0641

MAE: 3842.4093

For alpha = 0.5
Training set evaluation:
MSE: 24488680.1599
RMSE: 4948.6039
R2 Score: 0.0134
Adjusted R2 Score: -0.0017
MAE: 4003.0306

Test set evaluation:
MSE: 24278016.1557
RMSE: 4927.2727
R2 Score: 0.0000
Adjusted R2 Score: -0.0641
MAE: 3842.4093

For alpha = 1.0
Training set evaluation:
MSE: 24512862.9956
RMSE: 4951.0467
R2 Score: 0.0124
Adjusted R2 Score: -0.0027
MAE: 4001.7690

Test set evaluation:
MSE: 24278016.1557
RMSE: 4927.2727
R2 Score: 0.0000
Adjusted R2 Score: -0.0641
MAE: 3842.4093

For alpha = 2.0
Training set evaluation:

MSE: 24560175.8029

RMSE: 4955.8224

R2 Score: 0.0105

Adjusted R2 Score: -0.0046

MAE: 4001.7770

Test set evaluation:

MSE: 24278016.1557

RMSE: 4927.2727

R2 Score: 0.0000

Adjusted R2 Score: -0.0641

MAE: 3842.4093

MSE and RMSE metrics increase slightly as alpha increases, indicating that the model's errors are slightly higher with higher alpha values

The R2 Score is very low and close to zero for all alpha values, indicating that the model does not explain much of the variance in the target variable.

This metric decreases slightly as alpha increases, showing that the model's explanatory power decreases when accounting for the number of predictors.

The MAE is very similar across different alpha values, with a slight decrease as alpha increases

(h) (1.5 marks) Use the Gradient Boosting Regressor to perform regression on the preprocessed dataset from part (c). Report the evaluation metrics (MSE, RMSE, R2 score, Adjusted R2 score, MAE). Compare the results with those obtained in parts (c) and (g).

Gradient Boosting Regressor:

Training set evaluation:

MSE: 14926446.2573

RMSE: 3863.4759

R2 Score: 0.3986

Adjusted R2 Score: 0.3895

MAE: 3092.7482

Test set evaluation:

MSE: 24392500.9011

RMSE: 4938.8765

R2 Score: -0.0047

Adjusted R2 Score: -0.0691

MAE: 3815.7032

The Gradient Boosting Regressor performs better on the training set in terms of lower errors and better R2 scores, indicating it is better suited for capturing patterns in the training data.

The regularization models generally perform better on the test set, indicating they generalize better to unseen data compared to the Gradient Boosting Regressor.

The Gradient Boosting Regressor might be overfitting the training data, judging by performance on the test data.

The regularization models, not performing as well on the training set, perform better on the train data - seem to generalize better to new data.