CS5691: Pattern Recognition and Machine Learning
Assignment #1

Topics: K-Nearest Neighbours, Naive Bayes, Regression **Deadline:** 28 Feb 2023, 11:55 PM

Teammate 1: K.Saipranav Reddy **Roll number:** CS20B040
Teammate 2: M.D.Chakradhar **Roll number:** CS20B050

- Please refer to the **Additional Resources** tab on the Course webpage for basic programming instructions.
- This assignment has to be completed in teams of 2. Collaborations outside the team are strictly prohibited.
- Any kind of plagiarism will be dealt with severely. These include copying text or code from any online sources. These will lead to disciplinary actions according to institute guidelines. Acknowledge any and every resource used.
- Be precise with your explanations. Unnecessary verbosity will be penalized.
- Check the Moodle discussion forums regularly for updates regarding the assignment.
- You should submit a zip file titled **'rollnumber1_rollnumber2.zip'** on Moodle where rollnumber1 and rollnumber2 are your institute roll numbers. Your assignment will **NOT** be graded if it does not contain all of the following:
    1. Type your solutions in the provided LaTeX template file and title this file as **'Report.pdf'**. **State your respective contributions at the beginning of the report clearly.** Also, embed the result figures in your LaTeX solutions.
    2. Clearly name your source code for all the programs in **individual Google Colab files**. Please submit your code only as Google Colab file (.ipynb format). Also, embed the result figures in your Colab code files.
- We highly recommend using `Python 3.6+` and standard libraries like `NumPy, Matplotlib, Pandas, Seaborn`. Please use `Python 3.6+` as the only standard programming language to code your assignments. Please note: the TAs will only be able to assist you with doubts related to Python.
- You are expected to code all algorithms from scratch. **You cannot use standard inbuilt libraries for algorithms**. Using them will result in a straight zero on coding questions, `import` wisely!
- We have provided different training and testing sets for each team. f.e. train_1 and test_1 denotes training and testing set assigned to team id 1. Use sets assigned to your team only for all questions, reporting results using sets assigned to different team will result in straight zero marks.
- Any graph that you plot is unacceptable for grading unless it labels the x-axis and y-axis clearly.

- **Please start early and clear all doubts ASAP.**
- Please note that the TAs will **only** clarify doubts regarding problem statements. The TAs won't discuss any prospective solution or verify your solution or give hints.
- Please refer to the CS5691 PRML course handout for the late penalty instruction guidelines.
- Post your doubt only on Moodle so everyone is on the same page.

---

1. **[Regression]** You will implement linear regression as part of this question for the dataset1 provided here.

   Note that you can only regress over the points in the train dataset and you are not supposed to fit a curve on the test dataset. Whatever solution you get for the train data, you have to use that to make predictions on the test data and report results.

   (a) (2 marks) Use standard linear regression to get the best-fit curve. Split the data into train and validation sets and try to fit the model using a degree 1 polynomial then vary the degree term of the polynomial to arrive at an optimal solution.

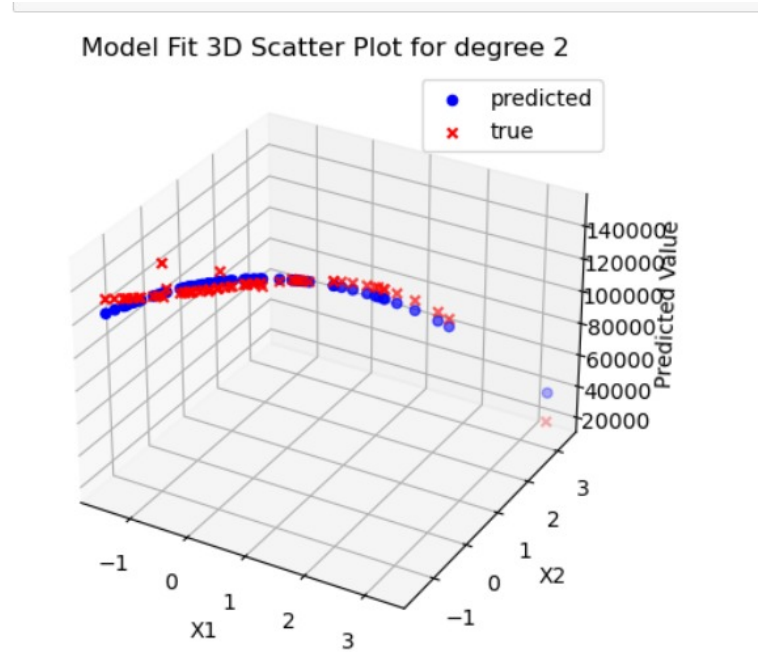
   For this, you are expected to report the following -

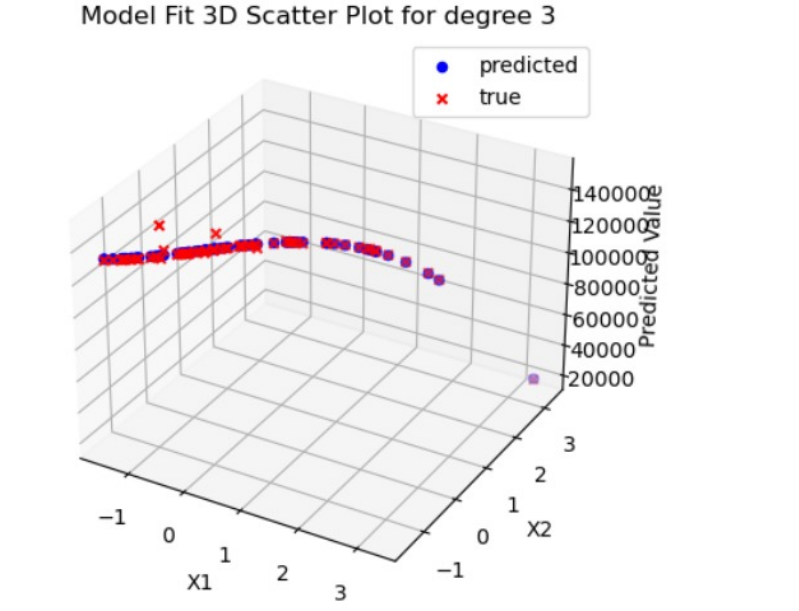
   - Plot different figures for train and validation data and for each figure plot curve of obtained function on data points for various degree term of the polynomial.(refer to fig. 1.4, Pattern Recognition and Machine Learning, by Christopher M. Bishop).
   - Plot the curve for Mean Square Error(MSE) Vs degree of the polynomial for train and validation data.(refer to fig. 1.5, Pattern Recognition and Machine Learning, by Christopher M. Bishop)
   - Report the error for the best model using Mean Square Error(MSE) for train and test data provided(Use closed-form solution).
   - Scatter plot of best model output vs expected output for both train and test data provided to you.
   - Report the observations from the obtained plots.

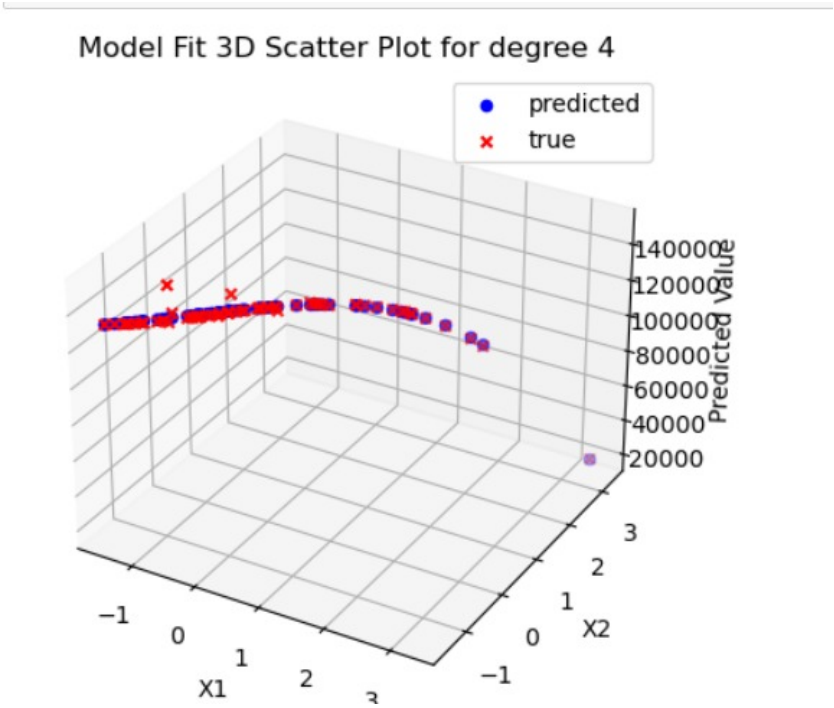
   Solution:

   We have split the given data into two parts
   i)Train Data (80%)
   ii)Validation data(20%)

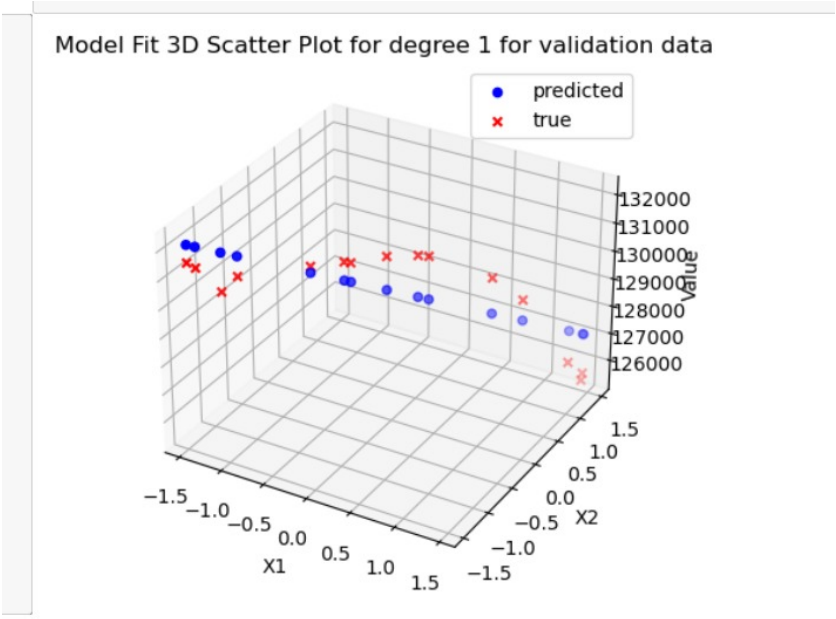
   Here are the plots for **train data** for various degrees of the polynomial.

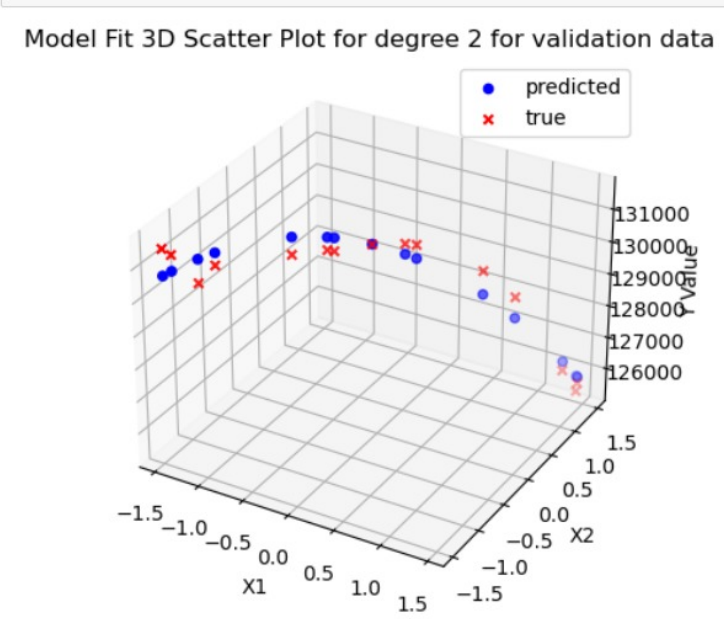

Model Fit 3D Scatter Plot for degree 2
predicted
true
Predicted Value
140000
120000
100000
80000
60000
40000
20000
X1
X2
−1
0
1
2
3

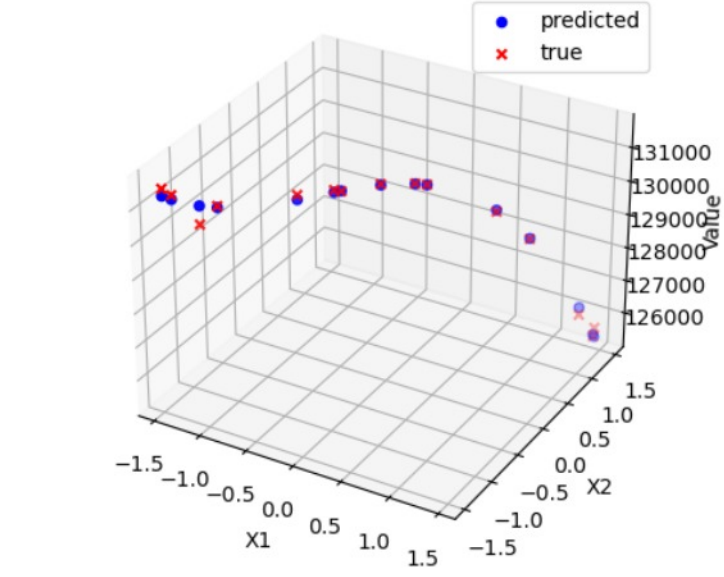

Model Fit 3D Scatter Plot for degree 3
predicted
true
Predicted Value
140000
120000
100000
80000
60000
40000
20000
X1
X2
−1
0
1
2
3

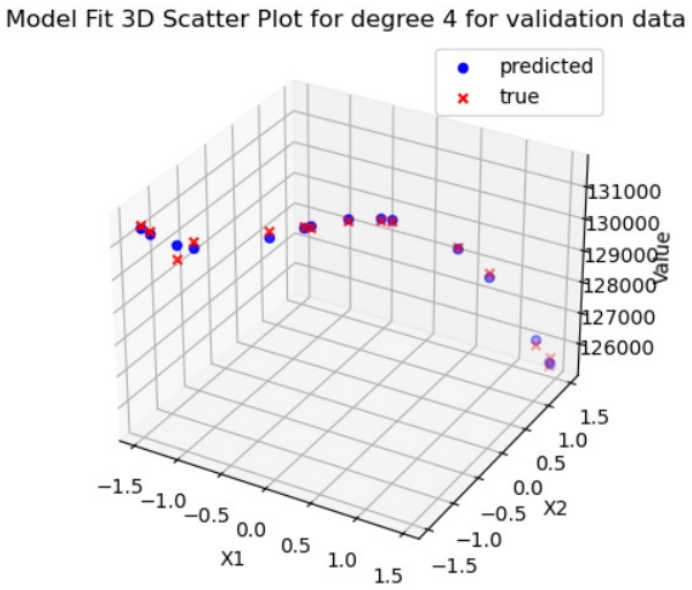


Here are the plots for **validation data** for various degrees of the polynomial.

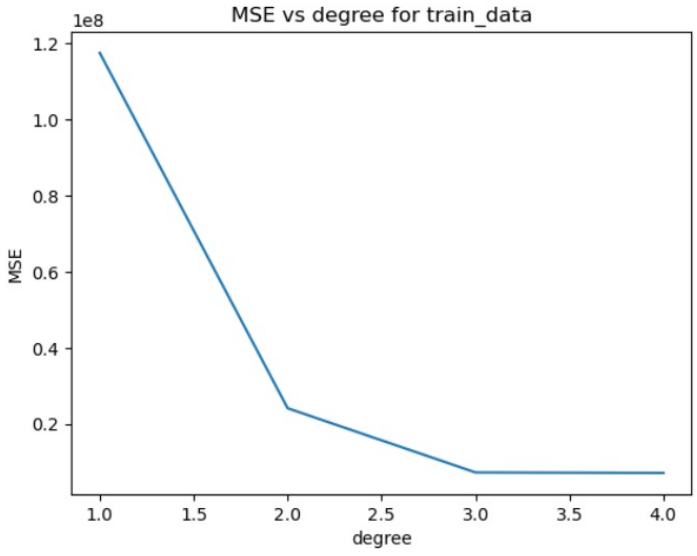


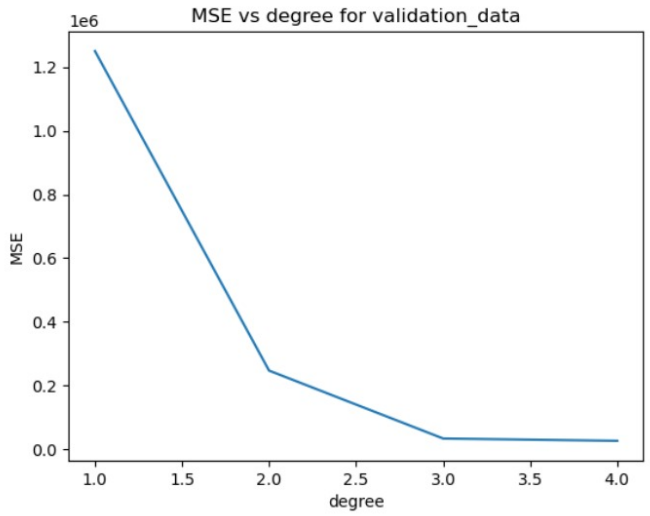

Model Fit 3D Scatter Plot for degree 2 for validation data
predicted
true
131000
130000
129000
128000
127000
126000
Value
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
X1
X2


Model Fit 3D Scatter Plot for degree 3 for validation data
predicted
true
131000
130000
129000
128000
127000
126000
Value
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
X1
X2



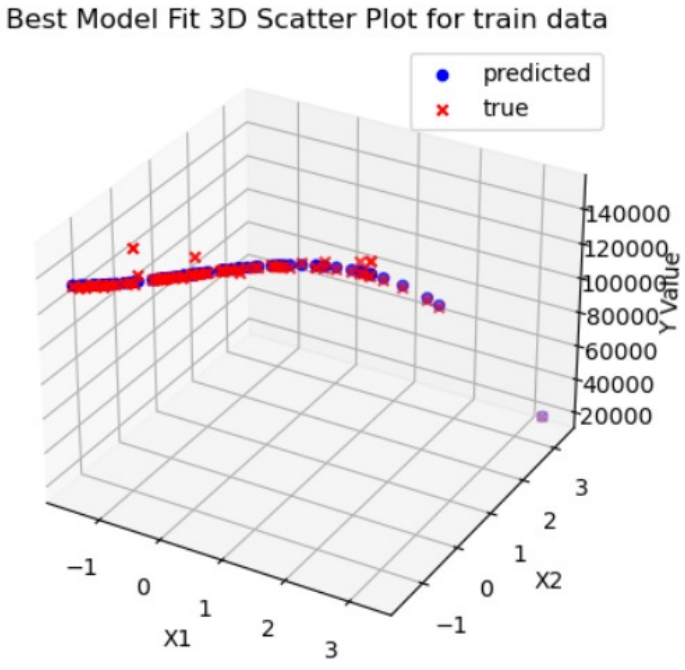
Here is the plot of **MSE** vs degree of polynomial for **train data** and **valid data**.

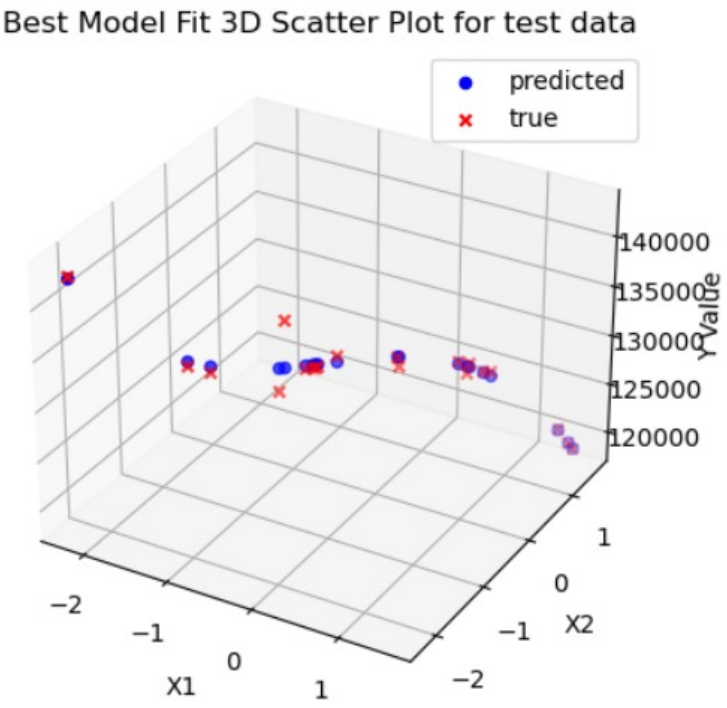




The best model is the polynomial with degree 3, the Mean square error for train and valid data sets are
MSE FOR TRAIN DATA = 7241671.642
MSE FOR TEST DATA = 1547743.1065

Here is the scatter plot of best model output vs expected output for train data and validation data.





Observation from the plots:

The best model is the polynomial of degree 3 , there is sudden decrease in MSE from degree 2 to degree 3. From degree 4 the graphs are overfitting.

(b) (3 marks) Split the data into train and validation sets and use ridge regression, then report for which value of lambda (λ) you obtain the best fit. For this, you are expected to report the following -

- Choose the degree from part (a), where the model overfits and try to control it using the regularization technique (Ridge regression).
- Use various choices of lambda(λ) and plot MSE test Vs lambda(λ).
- Report the error for the best model using Mean Square Error(MSE) for train and test data provided (Use closed-form solution).
- Scatter plot of best model output vs expected output for both train and test data provided to you.
- Report the observations from the obtained plots.

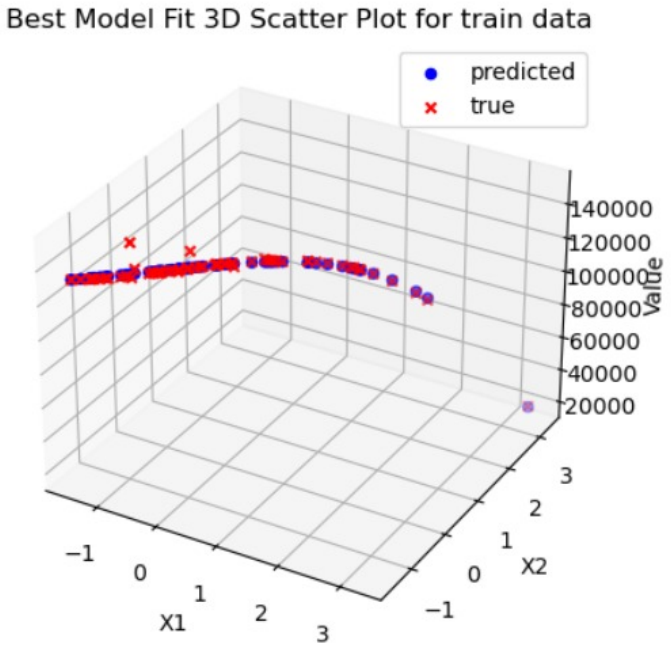
Solution:

Since we already found out that best degree is 3 , In this part we are choosing degree 4 which is overfit and finding the lambda for which we obtain the best fit using Ridge

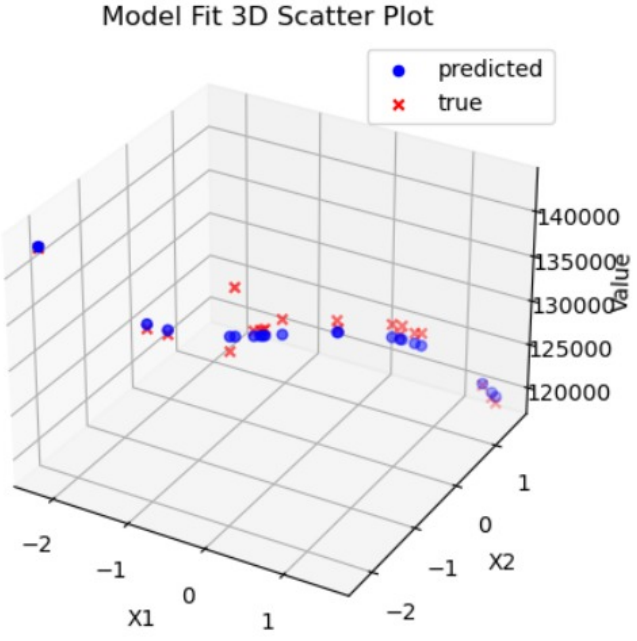
regression.

Scatter plots of best model for train and test data sets

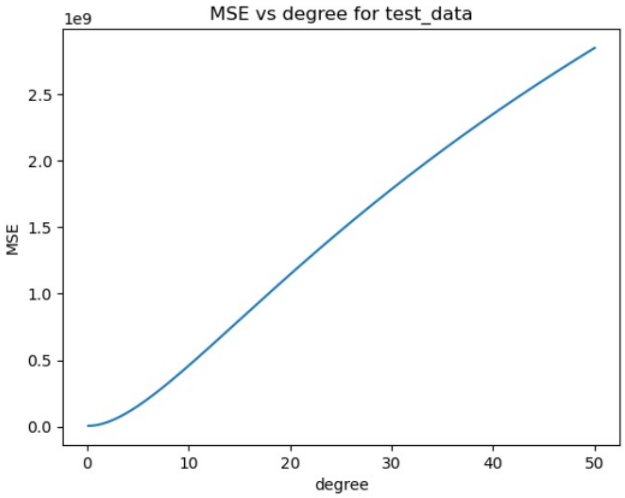
MSE FOR TRAIN DATA : 7107185.878



MSE FOR TEST DATA : 2397557.089



Here is plot of MSE vs degree for test



Observations: By looking at MSE vs Lambda plot we could see that error for lambda values 0.1 to 1 remains nearly the same and then we start noticing significant errors. This is underfitting .

When we first plotted the scatter plot of train data it kind of looked like a 2nd-degree curve and then we realized that when we plot it against degrees vs MSE then we found that for train data 3rd-degree gave the least error. So there is the problem of overfitting and then using this lambda method we can minimize the effect of degree and can reach a optimal solution.

2. **[Naive Bayes Classifier]** In this Question, you are supposed to build Naive Bayes classifiers for the datasets assigned to your team. Train and test datasets for each team can be found here. For each sub-question below, the report should include the following:

   - Accuracy on both train and test data.
   - Plot of the test data along with your classification boundary.
   - confusion matrices on both train and test data.

   You can refer to sample plots here and can refer Section 2.6 of "Pattern classification" book by [Duda et al. 2001] for theory.

   (a) (1 mark) Implement Naive Bayes classifier with covariance = I on dataset2. where, I denotes the identity matrix.

   Solution:

   Given : Train and test dataset2.

The Naive Bayes probability for more than one variables : The naive bayes probability is given by the product of

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

and probability of each class, where

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Here in this question,

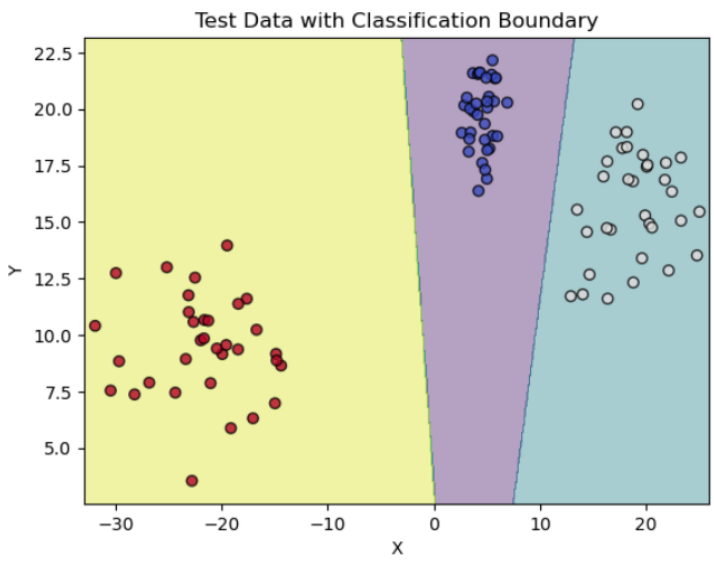
Σ is a 2 x 2 Identity Matrix.

Accuracy : Accuracy is given by the fraction of classes common between the given data and the obtained data.

We have found out the accuracy for both the test data and train data.
The accuracy for the train data : 0.998
The accuracy for the test data : 1.0

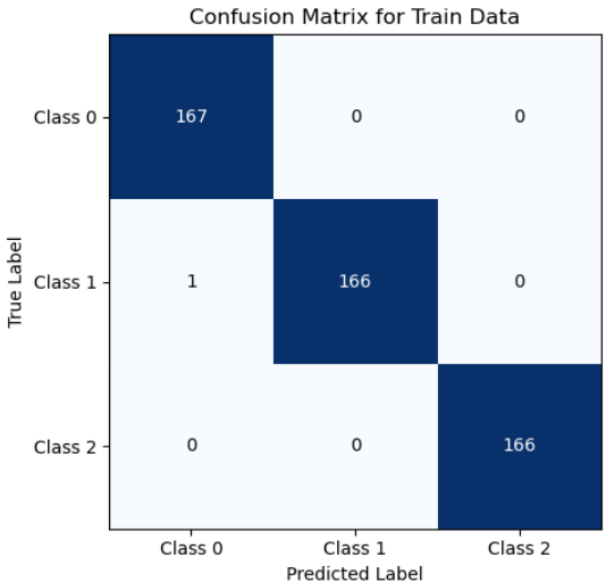
0.0.1 Classification Boundary

The plot of test data with the classification boundary :

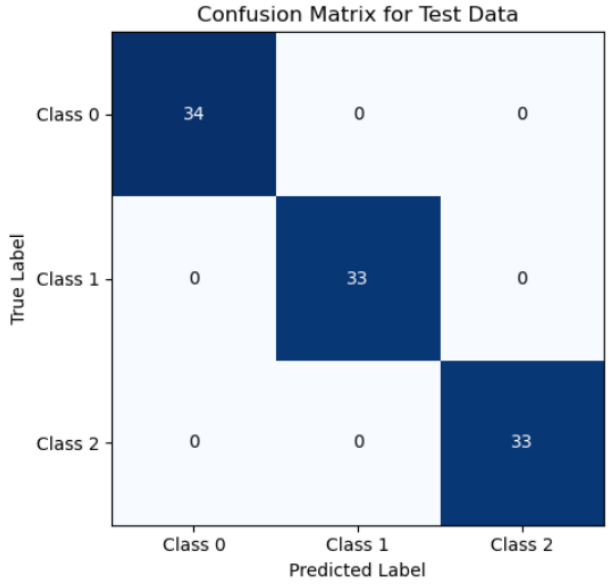


0.0.2 Confusion Matrices

The confusion matrix for train data :



The confusion matrix for test data :



(b) (1 mark) Implement Naive Bayes classifier with covariance = I on dataset3. where, I denotes the identity matrix.

Solution:

Given : Train and test dataset3.

The Naive Bayes probability for more than one variables : The naive bayes probability is given by the product of

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

and probability of each class, where

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Here in this question,

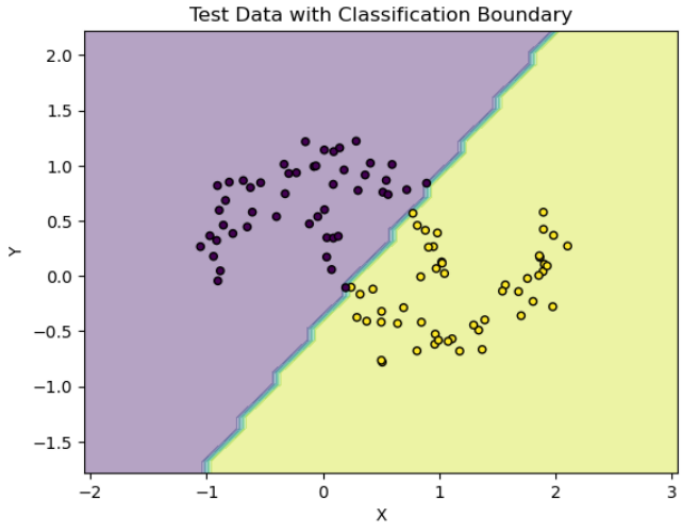
Σ is a 2 x 2 Identity Matrix.

Accuracy : Accuracy is given by the fraction of classes common between the given data and the obtained data.

We have found out the accuracy for both the test data and train data.
The accuracy for the train data : 0.786
The accuracy for the test data : 0.79

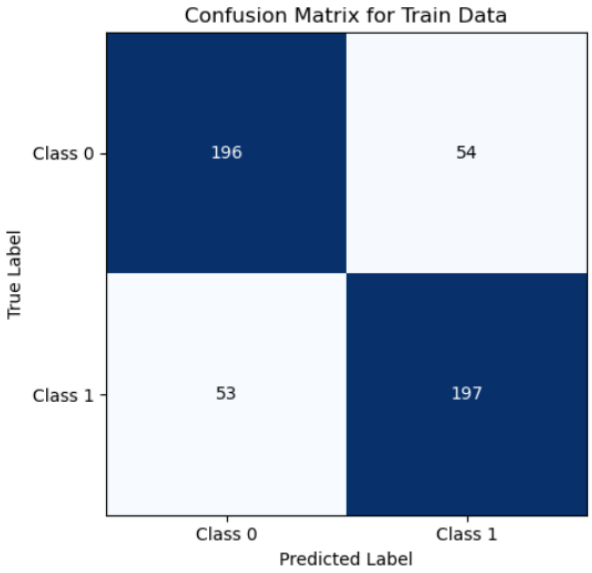
0.0.3 Classification Boundary

The plot of test data with the classification boundary :

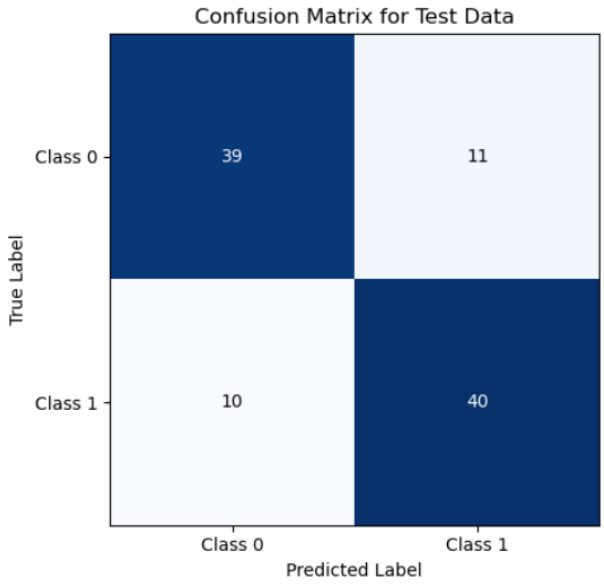


0.0.4 Confusion Matrices

The confusion matrix for train data :



The confusion matrix for test data :



(c) (1 mark) Implement Naive Bayes classifier with covariance same for all classes on dataset2.

Solution:

Given : Train and test dataset2.

.

The Naive Bayes probability for more than one variables : The naive bayes

probability is given by the product of

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

and probability of each class, where

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Here in this question,

$$\Sigma \text{ for train data}$$

is the covariance of the train data and is constant while dealing with the classes of train data.

$$\Sigma \text{ for test data}$$

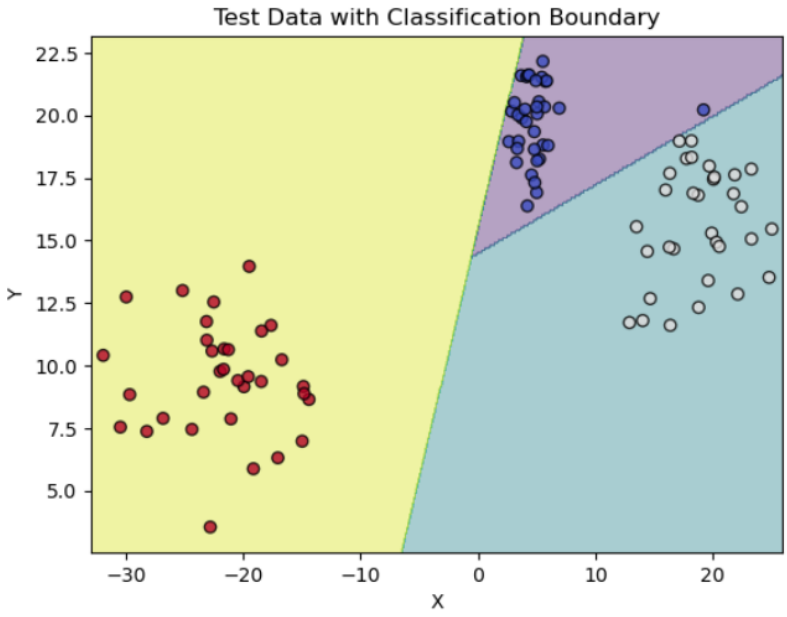
is the covariance of the test data and is constant while working with the classes of test data.

Accuracy : Accuracy is given by the fraction of classes common between the given data and the obtained data.

We have found out the accuracy for both the test data and train data.
The accuracy for the train data : 0.982
The accuracy for the test data : 0.99

0.0.5 Classification Boundary

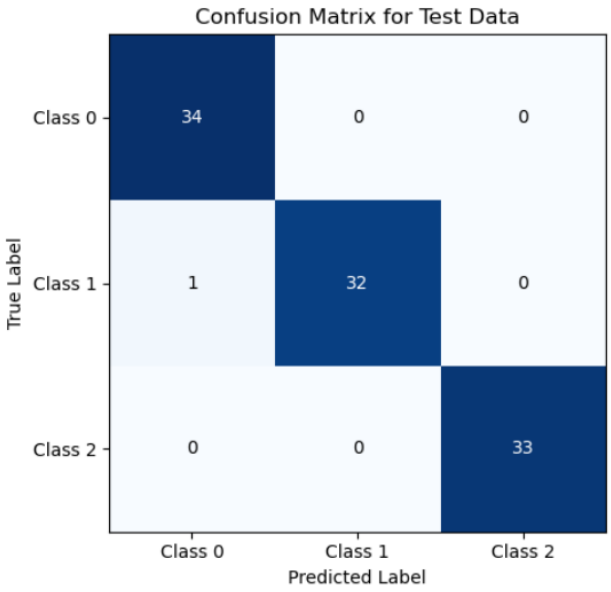
The plot of test data with the classification boundary :



0.0.6 Confusion Matrices

The confusion matrix for train data :

The confusion matrix for test data :



(d) (1 mark) Implement Naive Bayes classifier with covariance same for all classes on dataset3.

Solution:

Given : Train and test dataset3.

The Naive Bayes probability for more than one variables : The naive bayes probability is given by the product of

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

and probability of each class, where

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Here in this question,

$$\Sigma \text{ for train data}$$

is the covariance of the train data and is constant while dealing with the classes of train data.

$$\Sigma \text{ for test data}$$

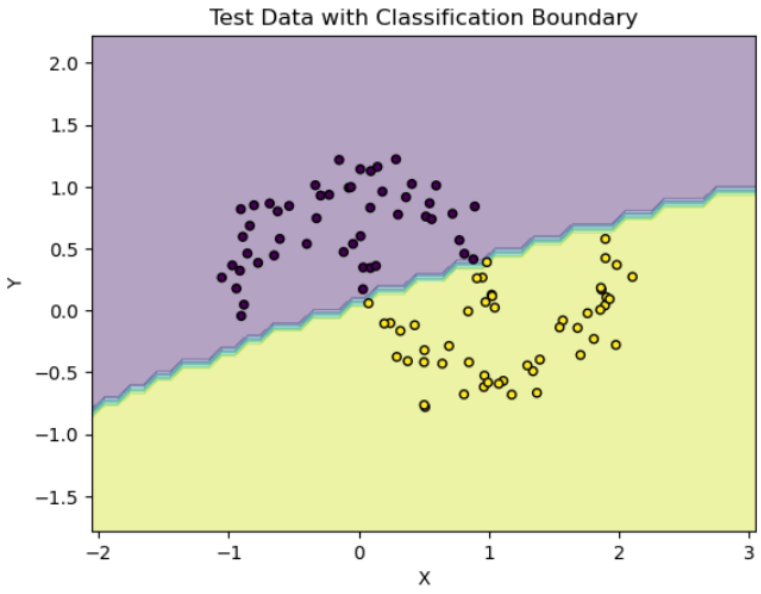
is the covariance of the test data and is constant while working with the classes of test data.

Accuracy : Accuracy is given by the fraction of classes common between the given data and the obtained data.

We have found out the accuracy for both the test data and train data.
The accuracy for the train data : 0.844
The accuracy for the test data : 0.84

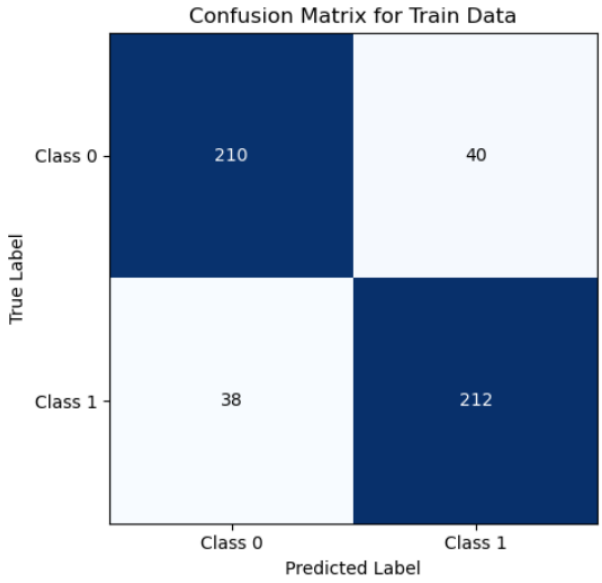
0.0.7 Classification Boundary

The plot of test data with the classification boundary :

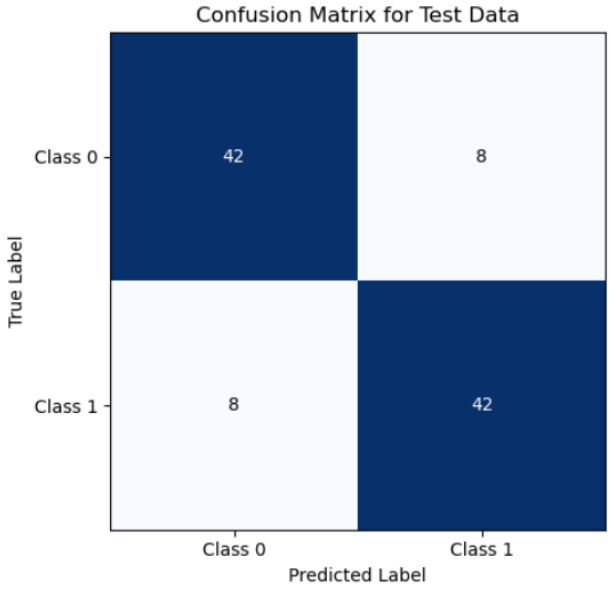


0.0.8 Confusion Matrices

The confusion matrix for train data :



The confusion matrix for test data :



(e) (1 mark) Implement Naive Bayes classifier with covariance different for all classes on dataset2.

Solution:

Given : Train and test dataset2.

The Naive Bayes probability for more than one variables : The naive bayes probability is given by the product of

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

and probability of each class, where

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Here in this question,

$$\Sigma \text{ for train data}$$

is the covariance, computed for each class and used separately while dealing with each class in train data.

$$\Sigma \text{ for test data}$$

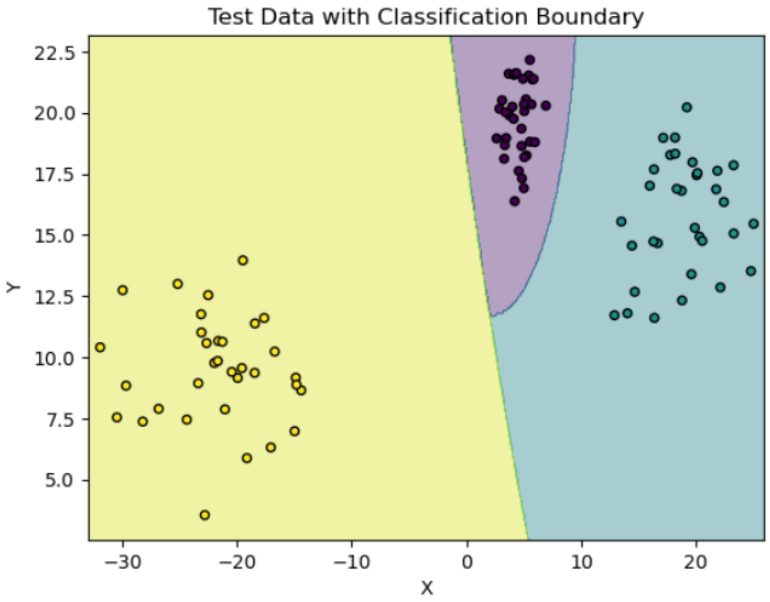
is the covariance, computed for each class and used separately while working with each class in test data.

Accuracy : Accuracy is given by the fraction of classes common between the given data and the obtained data.

We have found out the accuracy for both the test data and train data.
The accuracy for the train data : 1.0
The accuracy for the test data : 1.0

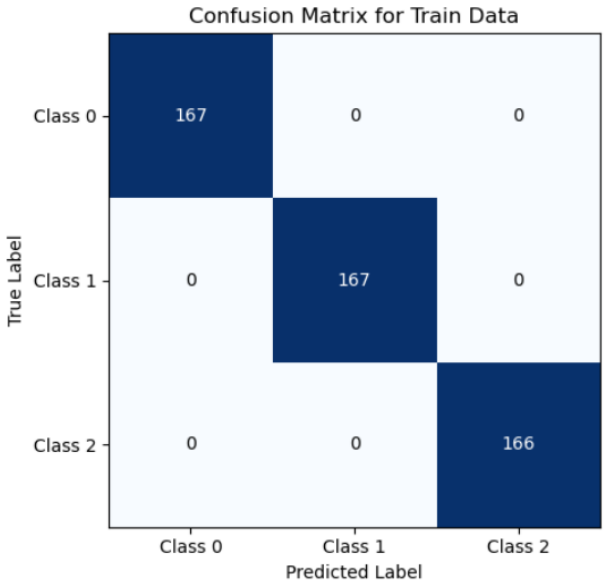
0.0.9 Classification Boundary

The plot of test data with the classification boundary :

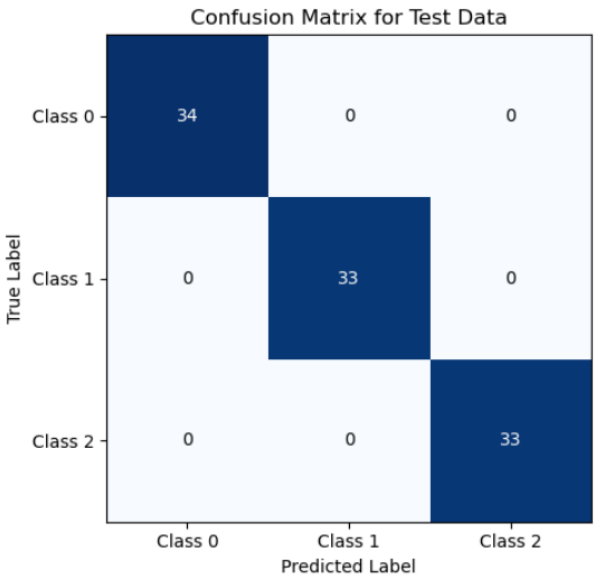


0.0.10 Confusion Matrices

The confusion matrix for train data :



The confusion matrix for test data :



(f) (1 mark) Implement Naive Bayes classifier with covariance different for all classes on dataset3.

Solution:

Given : Train and test dataset3.

The Naive Bayes probability for more than one variables : The naive bayes probability is given by the product of

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

and probability of each class, where

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Here in this question,

$$\Sigma \text{ for train data}$$

is the covariance, computed for each class and used separately while dealing with each class in train data.

$$\Sigma \text{ for test data}$$

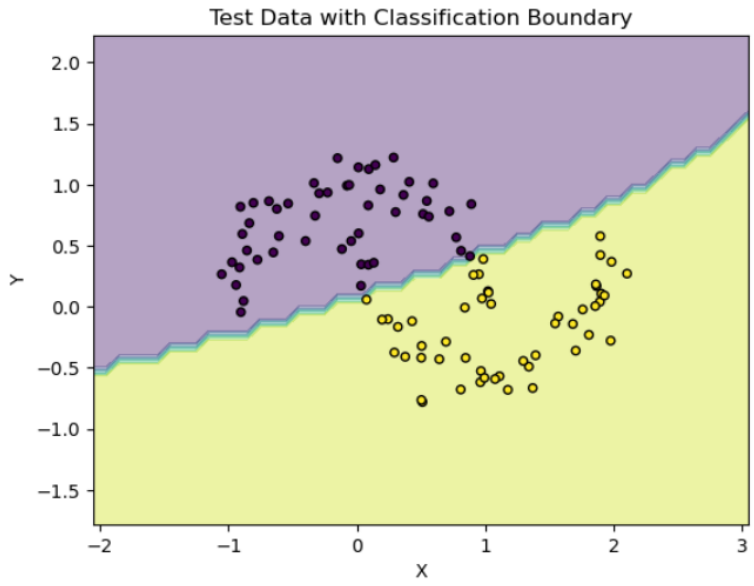
is the covariance, computed for each class and used separately while working with each class in test data.

Accuracy : Accuracy is given by the fraction of classes common between the given data and the obtained data.

We have found out the accuracy for both the test data and train data.
The accuracy for the train data : 0.842
The accuracy for the test data : 0.84

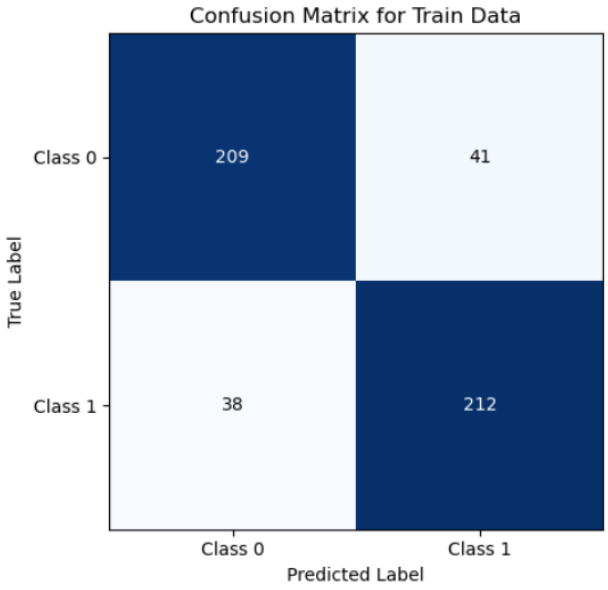
0.0.11 Classification Boundary

The plot of test data with the classification boundary :

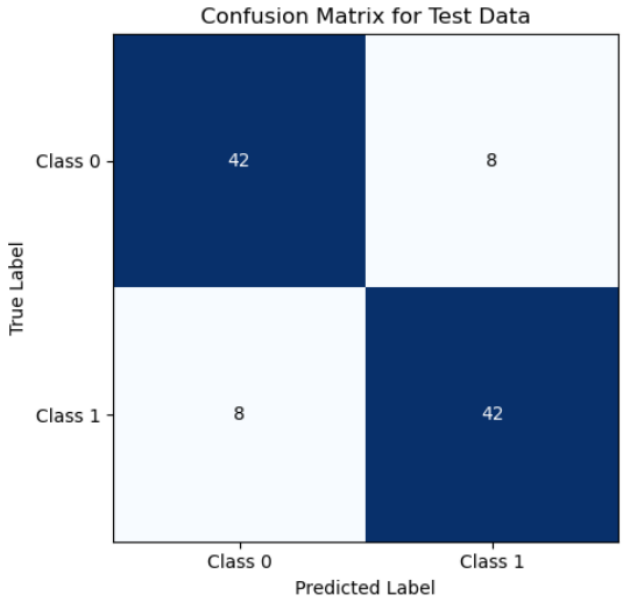


0.0.12 Confusion Matrices

The confusion matrix for train data :



The confusion matrix for test data :



3. **[KNN Classifier]** In this Question, you are supposed to build the k-nearest neighbors classifiers on the datasets assigned to your team. Dataset for each team can be found here. For each sub-question below, the report should include the following:

   - Analysis of classifier with different values of k (number of neighbors).
   - Accuracy on both train and test data for the best model.
   - Plot of the test data along with your classification boundary for the best model.

- confusion matrices on both train and test data for the best model.

(a) (2 marks) Implement k-nearest neighbors classifier on dataset2.

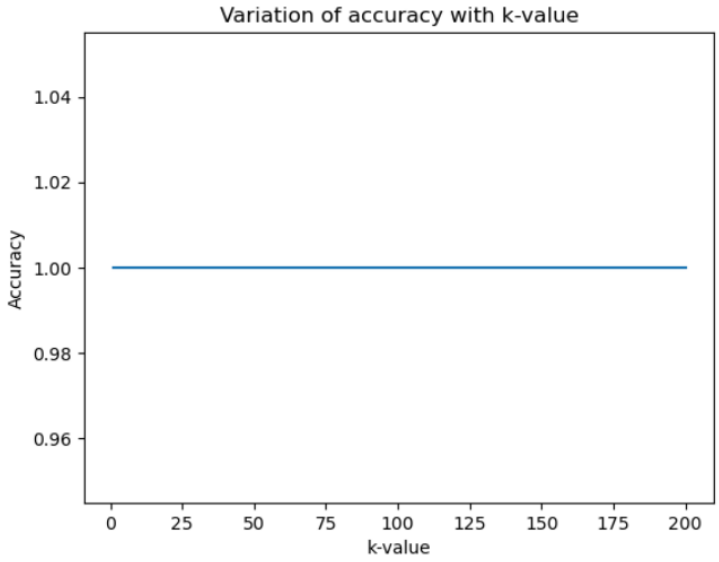
Solution:

Given : Train and Test dataset2.
Train data is divided into train (75 %) and validation data (25 %)

The KNN function : finding the distances sorting them out, collecting the first k nearest neighbours and finding the label with max count in each iteration and returning the list of labels with respective max counts.

The accuracy function : finding the number of labels common between the obtained list and the given list.

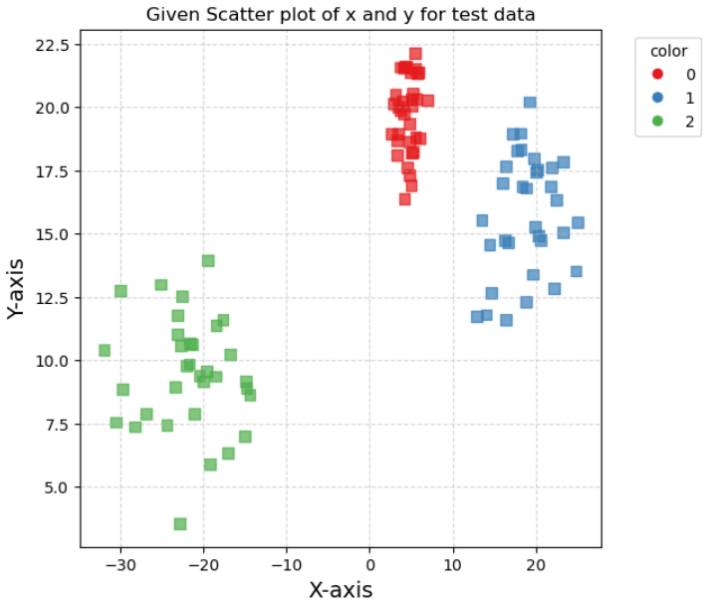
Then, we have checked the variations in accuracy on changing the values of k. The respective table of k-value and accuracy is uploaded in 'TrainDataSet2Accuracy.csv' file and the graph between k and accuracy looks like this :



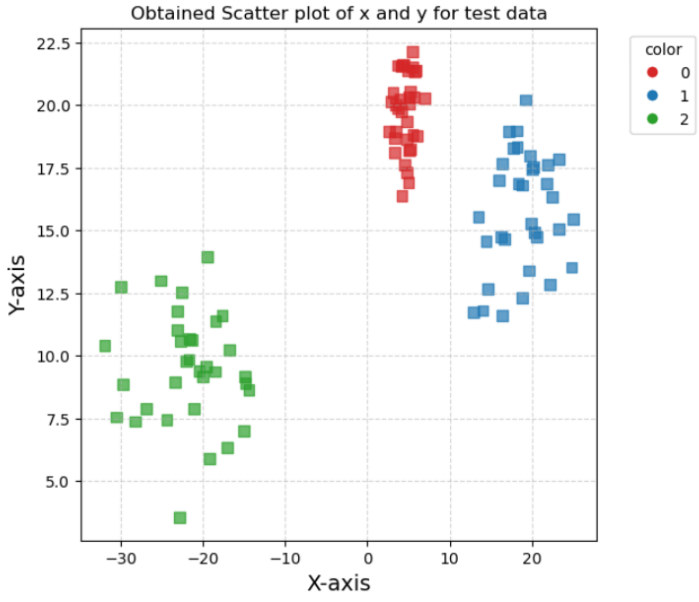
Based on the obtained data we have k=1 as the best model.
Accuracy of train data for best model (k=1) : 1.0
Accuracy of test data for best model (k=1) : 1.0

0.0.13 Scatter Plots

The scatter plot for given test data is :

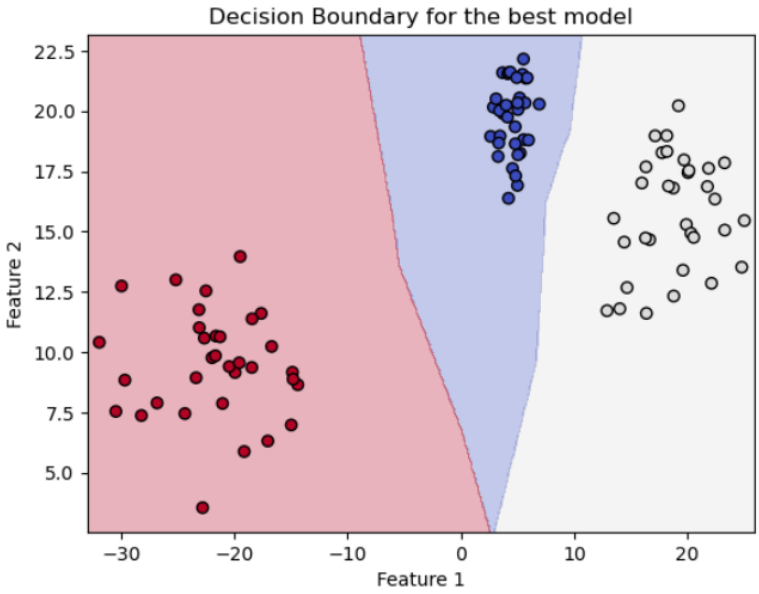


The scatter plot for obtained test data is :



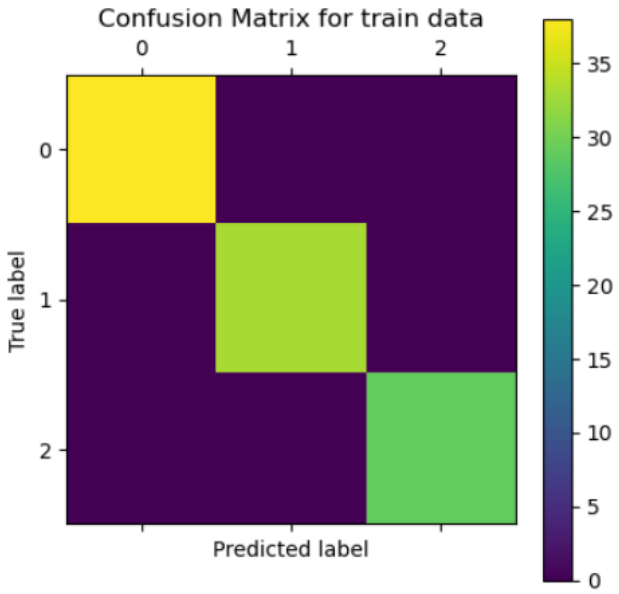
0.0.14 Classification boundary

The classification boundary for the predicted test data is :



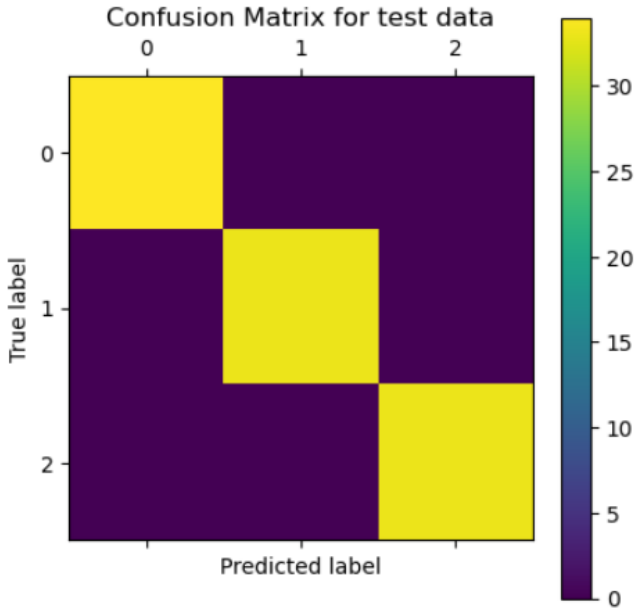
0.0.15 Confusion Matrix

The confusion matrix for train data is :




```
[[38.  0.  0.]
 [ 0. 33.  0.]
 [ 0.  0. 29.]]
```

The confusion matrix for test data is :




```
[[34.  0.  0.]
 [ 0. 33.  0.]
 [ 0.  0. 33.]]
```

(b) (2 marks) Implement k-nearest neighbors classifier on dataset3.

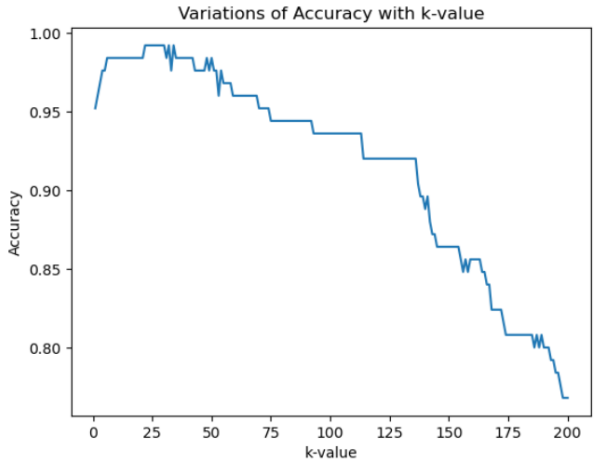
Solution:

Given : Train and Test dataset3
Train data is divided into train (75 %) and validation data (25 %)

The KNN function : finding the distances sorting them out, collecting the first k nearest neighbours and finding the label with max count in each iteration and returning the list of labels with respective max counts.

The accuracy function : finding the number of labels common between the obtained list and the given list.

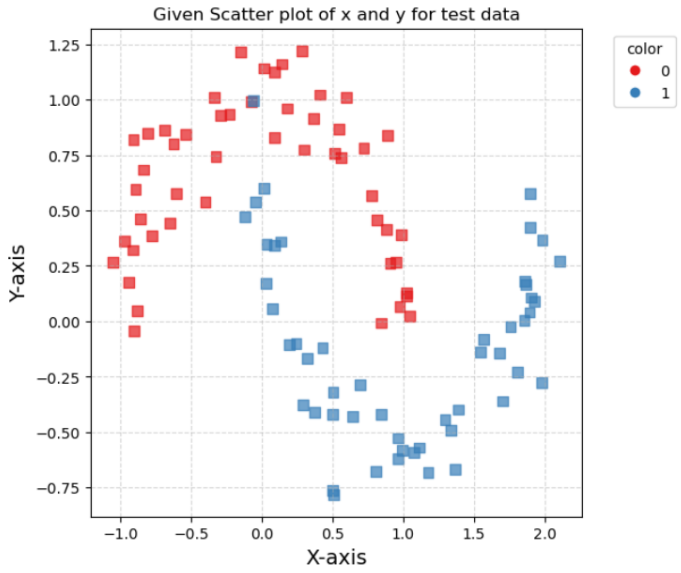
Then, we have checked the variations in accuracy on changing the values of k. The respective table of k-value and accuracy is uploaded in 'TrainDataSet3Accuracy.csv' file and the graph between k and accuracy looks like this :



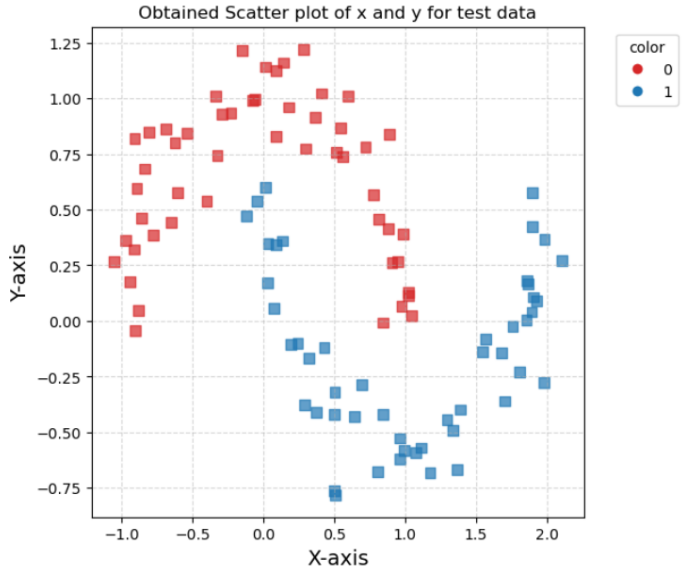
Based on the obtained data we have k=23 as the best model.
Accuracy of train data for best model (k=1) : 0.992
Accuracy of test data for best model (k=1) : 0.99

0.0.16 Scatter Plots

The scatter plot for given test data is :

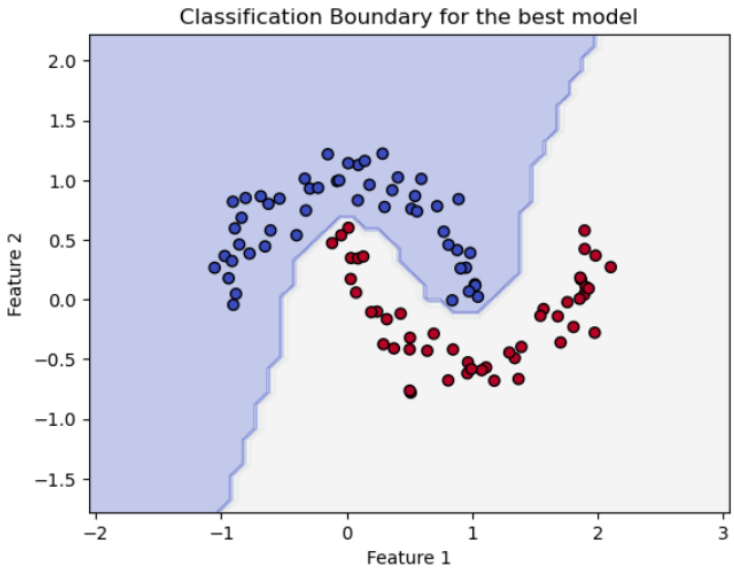


The scatter plot for obtained test data is :



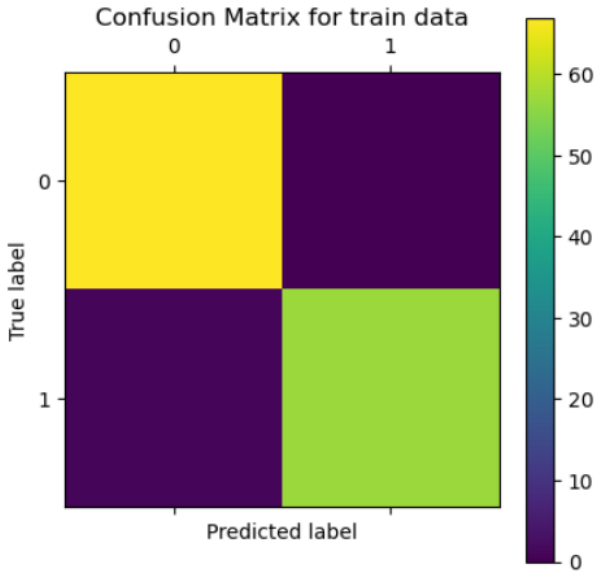
0.0.17 Classification boundary

The classification boundary for the predicted test data is :



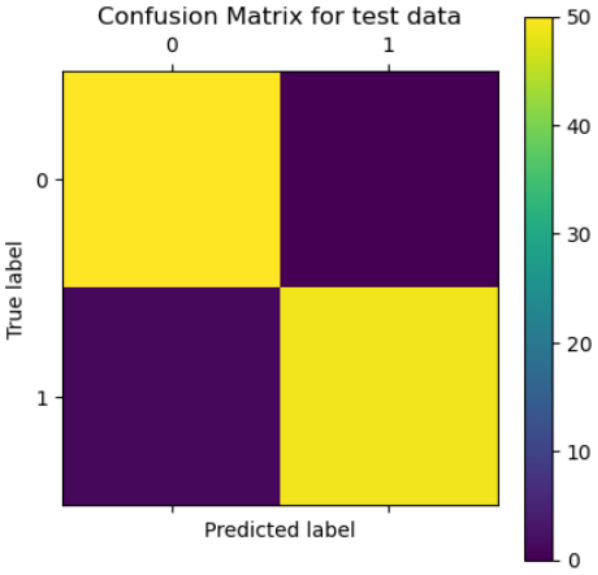
0.0.18 Confusion Matrix

The confusion matrix for train data is :




```
[[67.  0.]
 [ 1. 57.]]
```

The confusion matrix for test data is :




```
[[50.  0.]
 [ 1. 49.]]
```