CS5691: Pattern Recognition and Machine Learning
Assignment #1

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

Teammate 1: P Ayyappa Kethan **Roll number:** CS20B058
Teammate 2: Prakash A **Roll number:** CS20B061

Contributions: P Ayyappa Kethan : Q1,Q3(code) , Prakash A : Q2,Q3(report)

- 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.

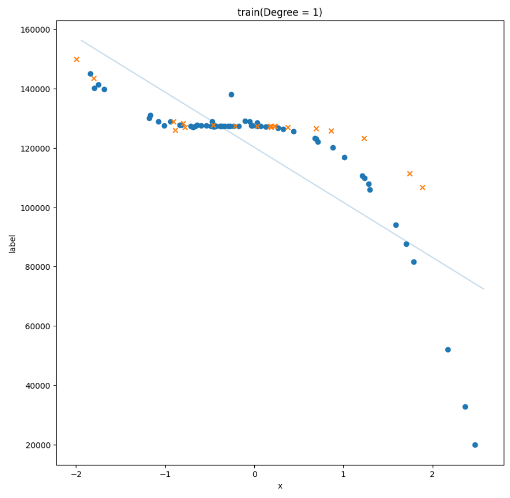
   For this, you are expected to report the following -

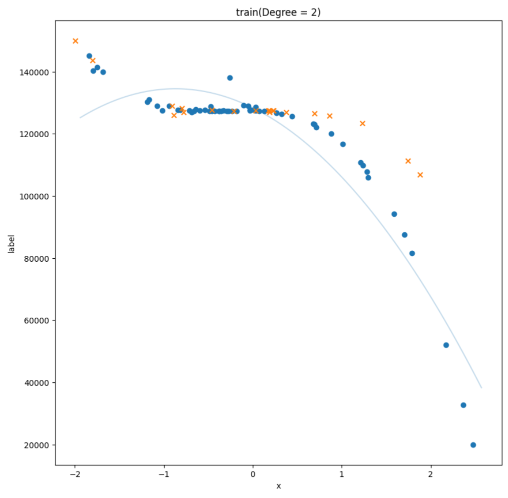
   - Plot different figures for train and validation data and for each figure plot curve of obtained function on data points for various degree term of the polynomial.(refer to fig. 1.4, Pattern Recognition and Machine Learning, by Christopher M. Bishop).
   - Plot the curve for Mean Square Error(MSE) Vs 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.

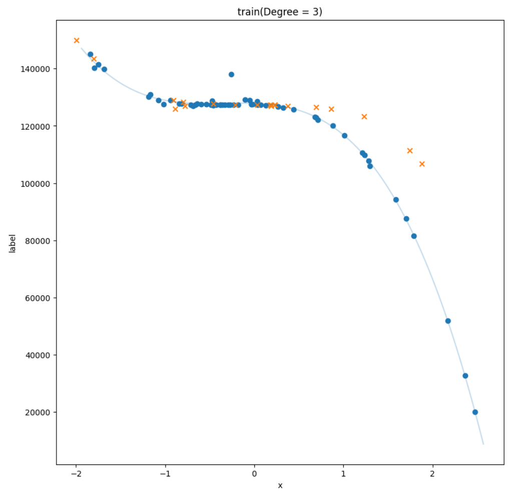
   Solution:

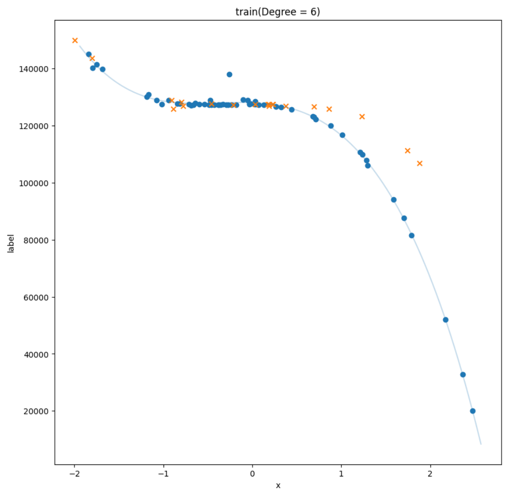
   - The regression is done based on taking loss function as $\Sigma(\mathrm{y}_i\text{-f}(\mathrm{x}_i))^2$
   - Training Split - 0.75, Test Split - 0.25
   - Z-Score Normalization is used
   - We dropped one feature column because correlation between 1st and 2nd column is 100 percent

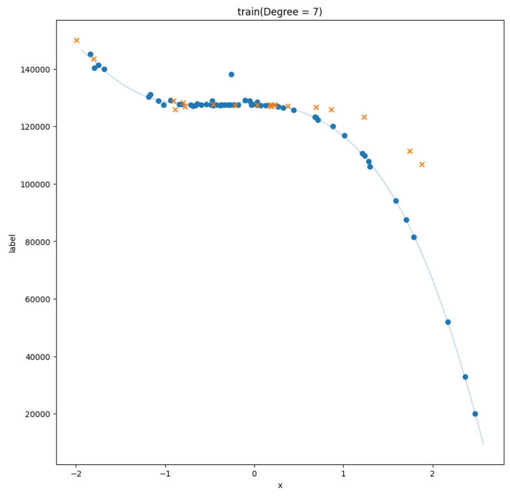
Model Curves for various degrees:

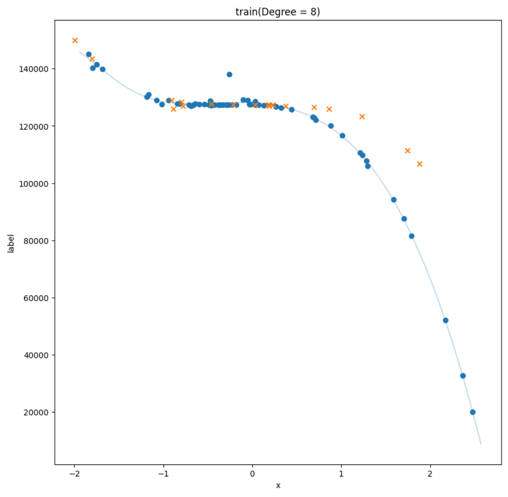




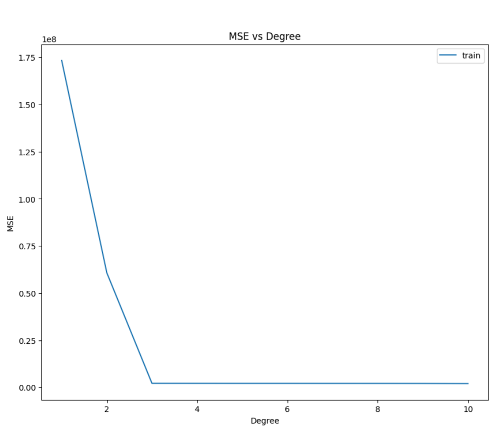


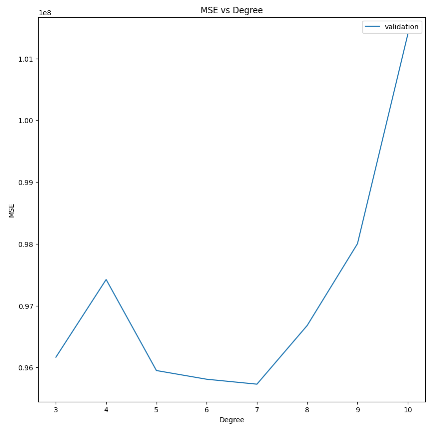






The plot for **mean square error vs degree** for train and validation data can be found below

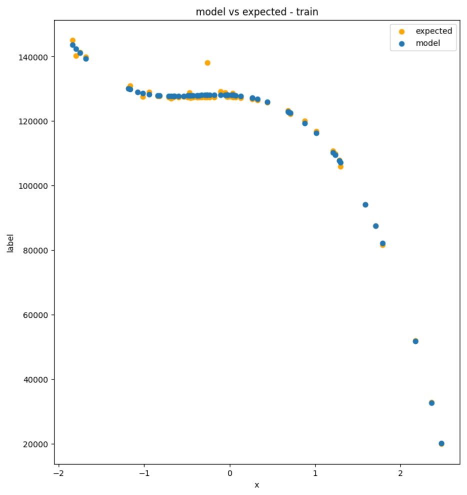


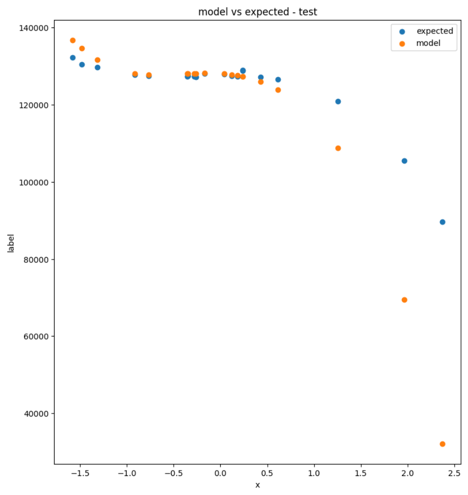


The best model is observed at **degree = 7**

The MSE for best model are reported below.
MSE for train : 2108033.27
MSE for test : 239853804.98

Scatterplot





Observations:

- MSE vs Degree for validation data given degree = 7 as best value
- The MSE for train data is decreasing continuously as expected
- Degree 6,7,8 have almost MSE values and graphs look similar

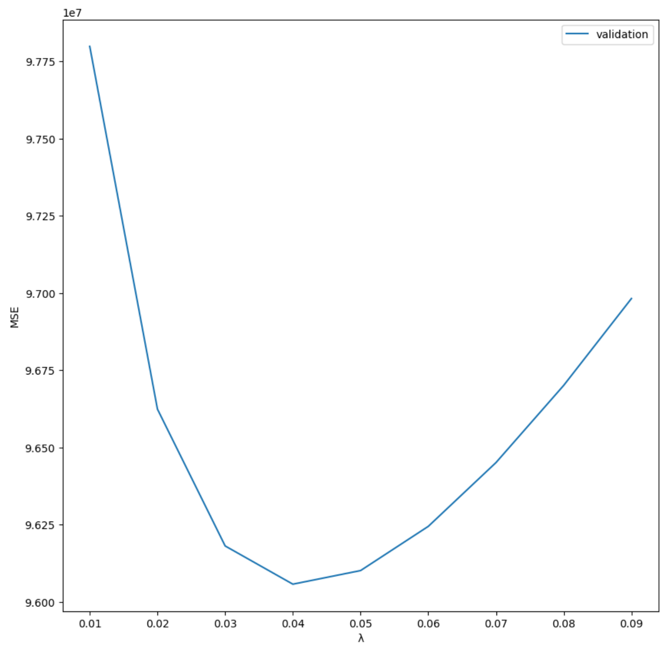
(b) (3 marks) Split the data into train and validation sets and use ridge regression, then report for which value of lambda (λ) you obtain the best fit. For this, you are expected to report the following -

- 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.

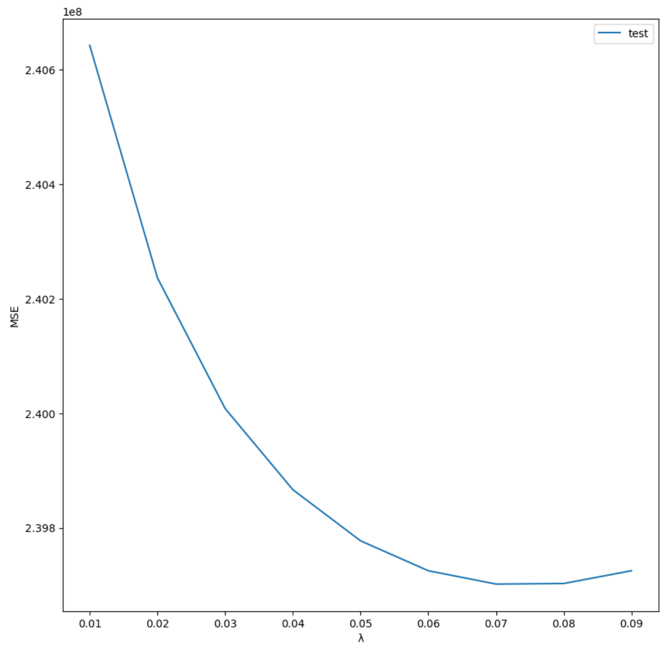
Solution:

- The regression is done based on taking loss function as $(\Sigma(y_i - f(x_i))^2 + \Sigma\ \beta_i{}^2)$
- Training Split - 0.75, Test Split - 0.25
- Z-Score Normalization is used
- We dropped one feature column because correlation between 1st and 2nd column is 100 percent
- The degree chosen for regularization is 10

MSE vs log(λ) on validation data is given below:

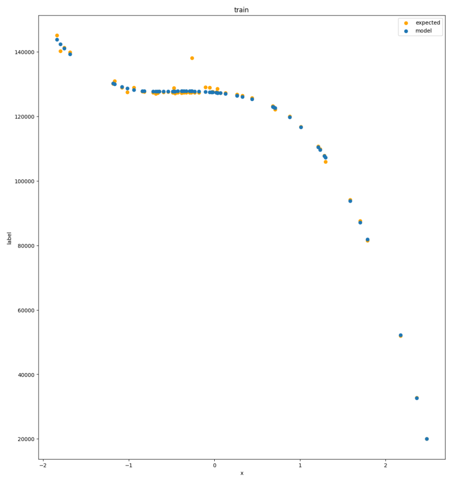


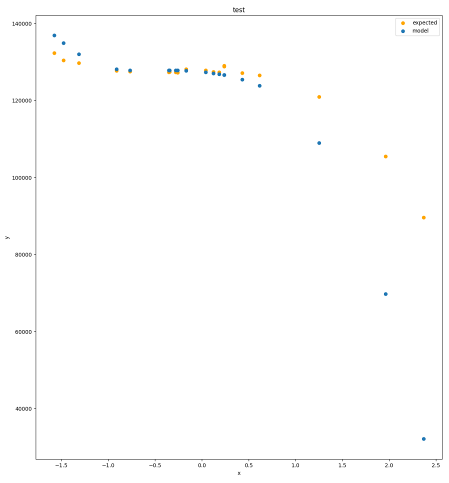
MSE vs lambda for test data is given below



The MSE for best model are reported below.
MSE for train : 2166274.32
MSE for test : 239866436.82

Scatterplots for train and test respectively :





Observations:

- We can see in MSE vs $\log(\lambda)$ with increase in lambda MSE increses.
- The best value for MSE is observed at $\mathbf{\log(\lambda) = -4}$
- MSE for test data is large despite being the best model

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.

   Code for below questions can be found here

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

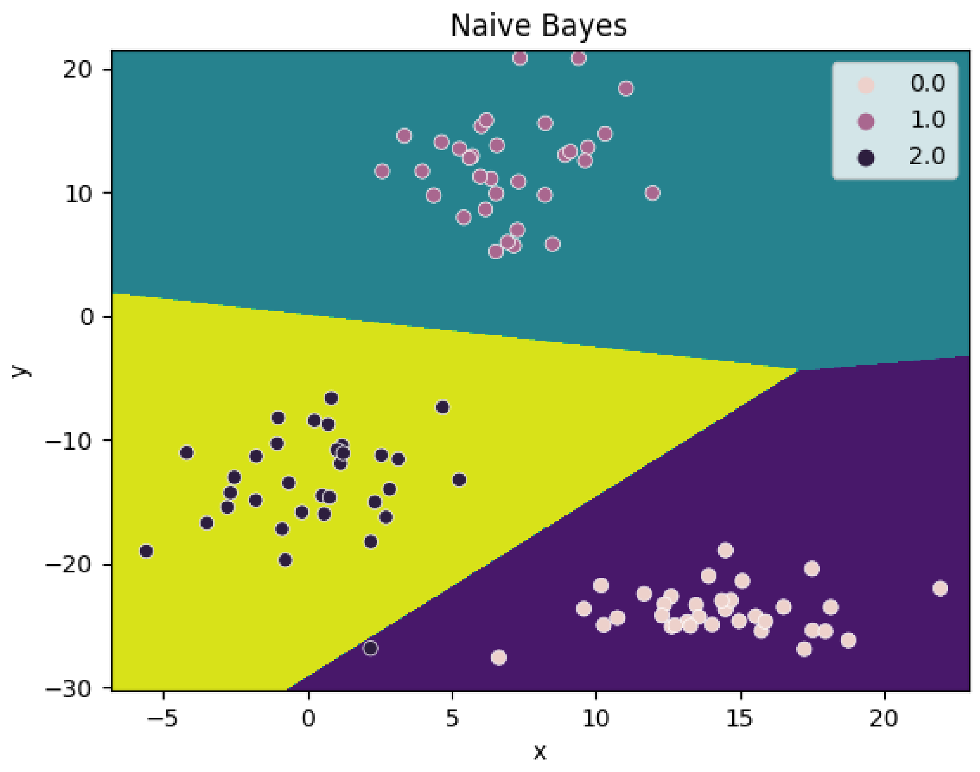
   Solution:

   In dataset2 we have 2 features and 3 classifications

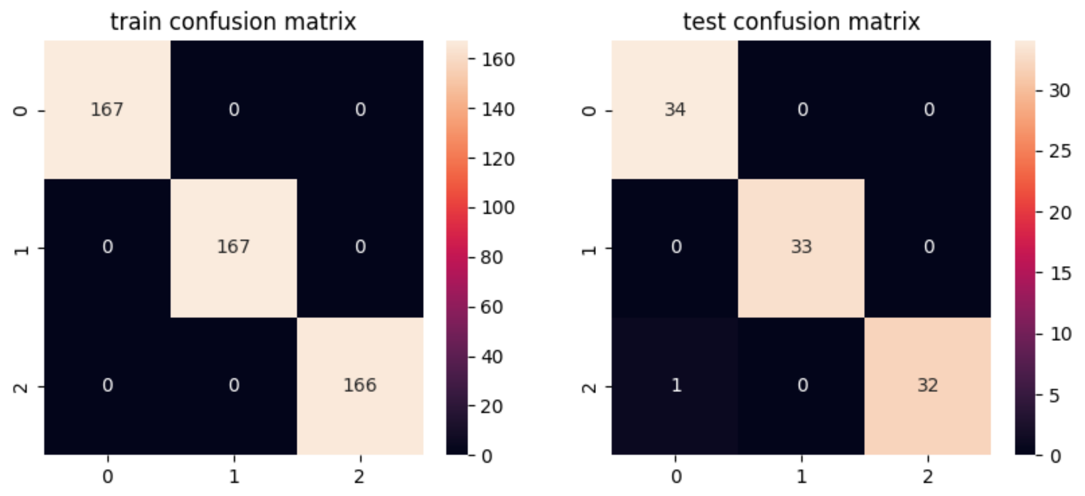
   In this Question the covariance matrix given is I (Identity matrix) which may not be the most suitable matrix for getting better accuracy and we will be using the same covariance matrix for all the classes in building the Gaussian Distribution function for each class.

   Accuracy for this model : Test Accuracy = 99 Train Accuracy = 100

   Plot for this model:



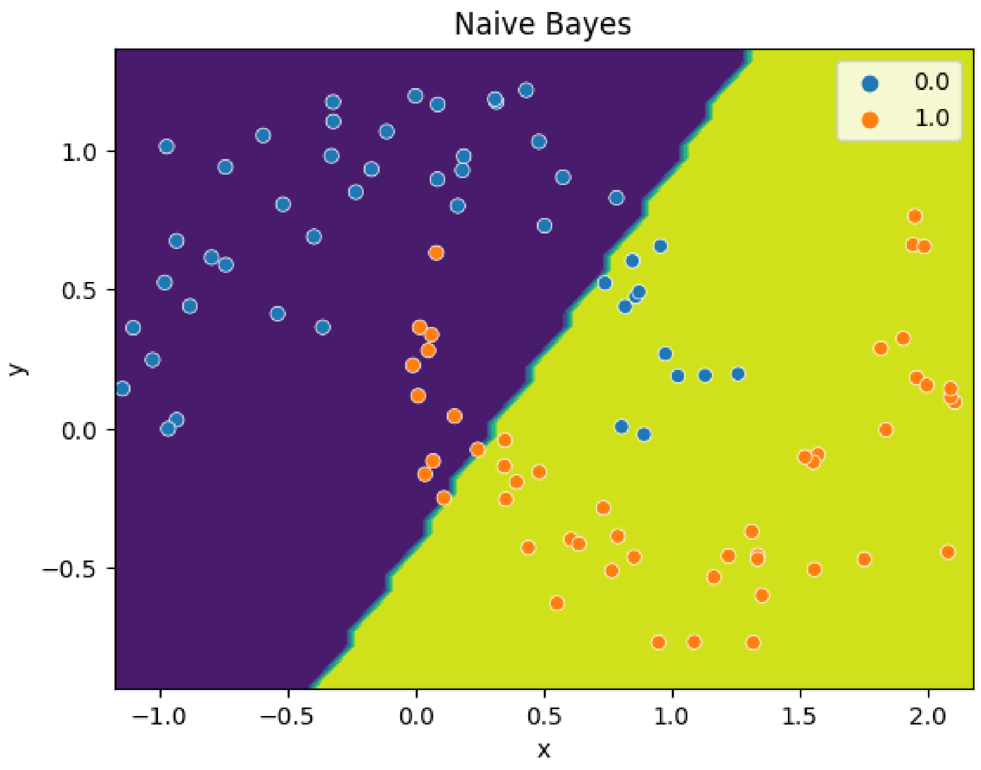
Confusion Matrix:



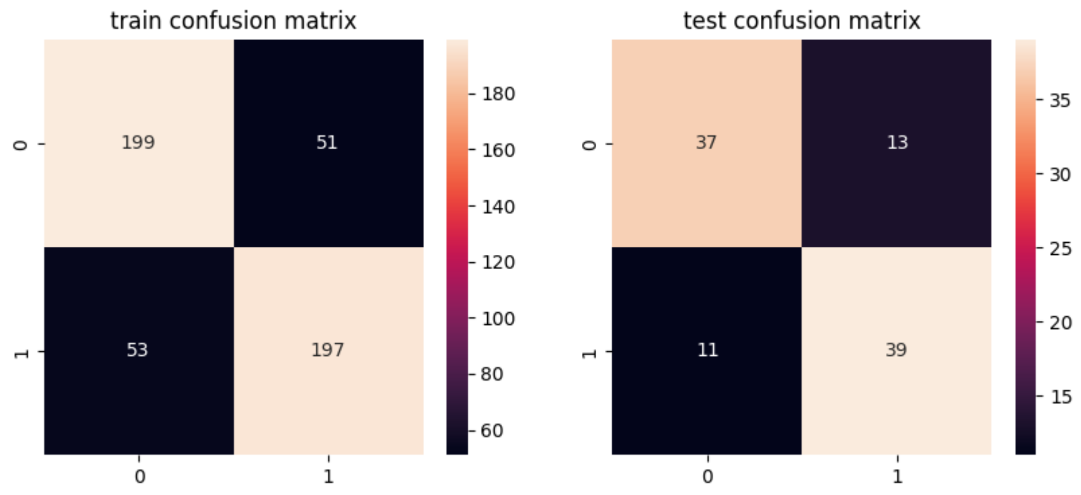
(b) (1 mark) Implement Naive Bayes classifier with covariance = I on dataset3. where, I denotes the identity matrix.

Solution:

In dataset3 we have 2 features and 2 classifications.
Accuracy for this model : Test Accuracy = 76 Train Accuracy = 79.2
Plot for this model:



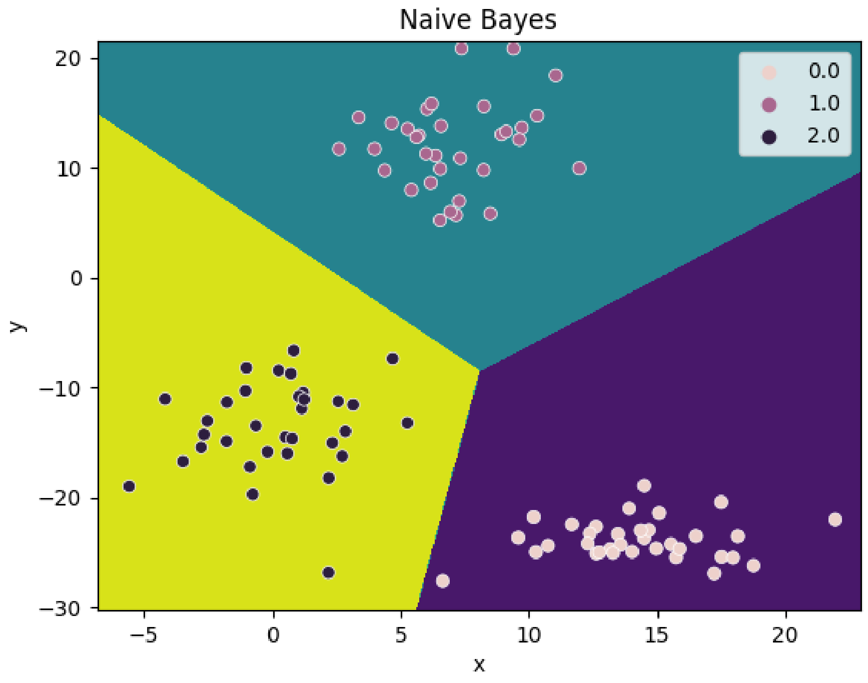
Confusion Matrix:



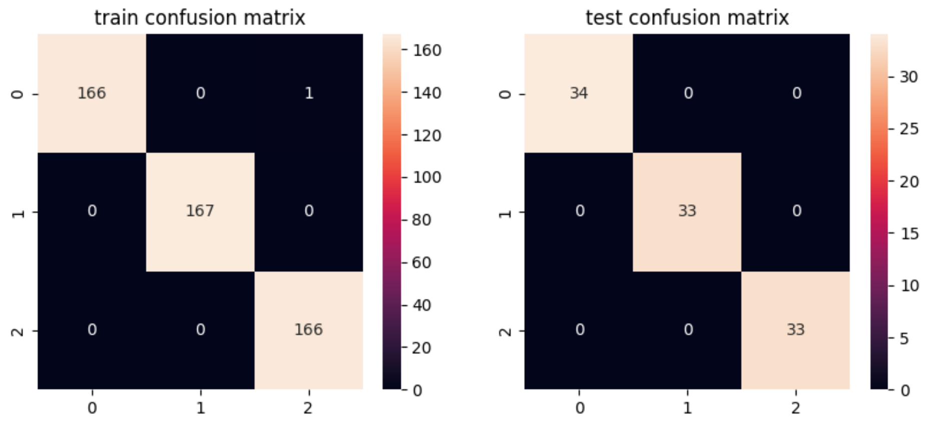
(c) (1 mark) Implement Naive Bayes classifier with covariance same for all classes on dataset2.

Solution:
The covariance matrix used in code is found using the data given for the 2 features. since covariance matrix is diagonal matrix (because we assumed the features (x1,x2) to be independent of each other cov(x1,x2) = 0). covariance matrix = [[var(x1),0][0,var(x2)]] (var(x1) = variance of x1 feature (all the values of x1 are used irrespective class))

Accuracy for this model : Test Accuracy = 100 Train Accuracy = 99.8
Plot for this model:



Confusion Matrix:



	0	1	2
0	166	0	1
1	0	167	0
2	0	0	166

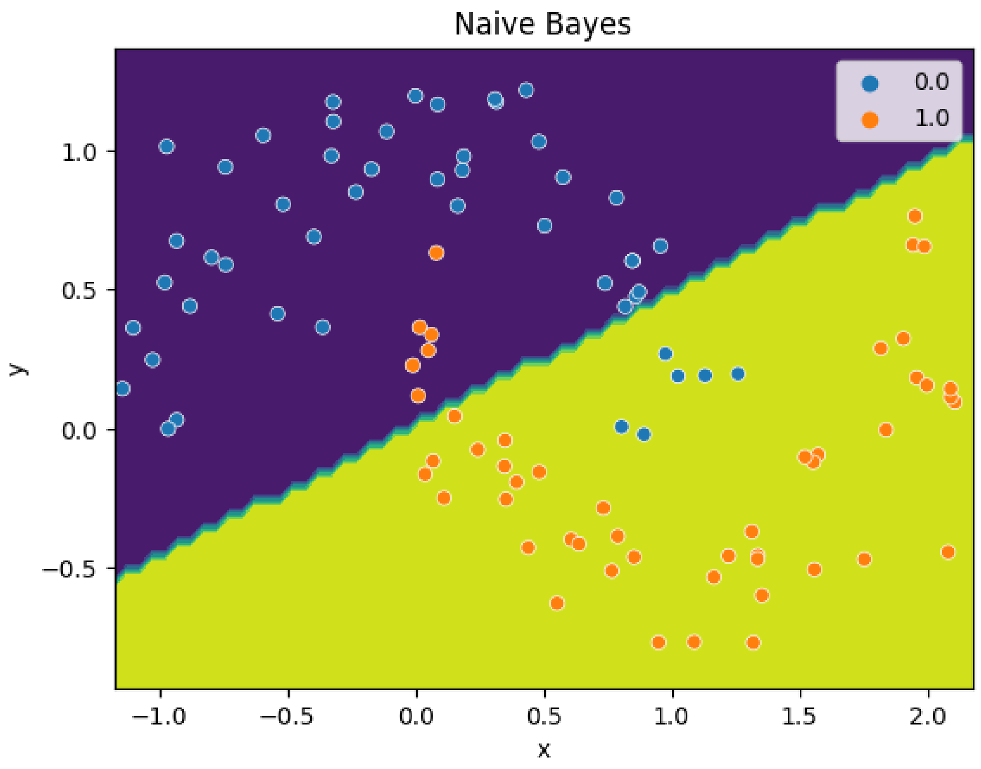
	0	1	2
0	34	0	0
1	0	33	0
2	0	0	33

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

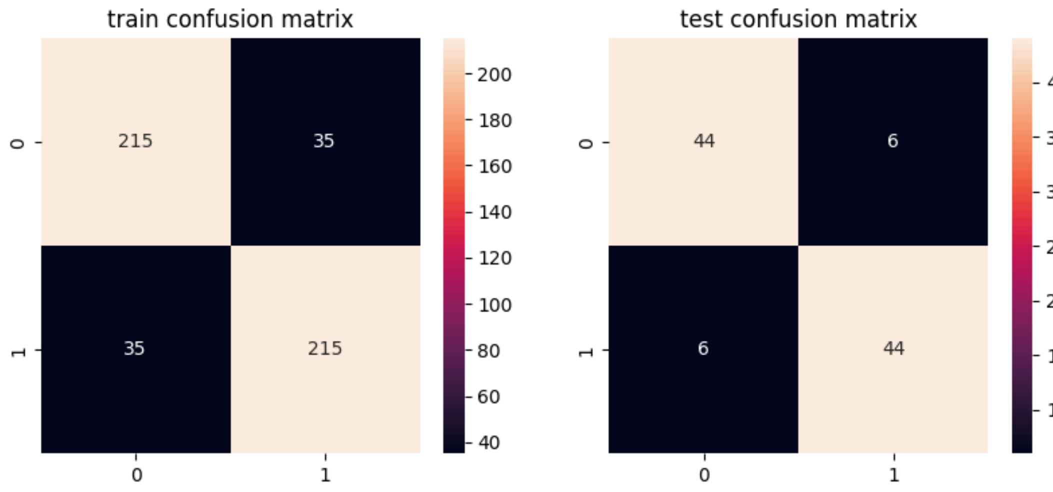
Solution: The covariance used in code is found using similar method as we did in part (c)

Accuracy for this model : Test Accuracy = 88 Train Accuracy = 86

Plot for this model:



Confusion Matrix:



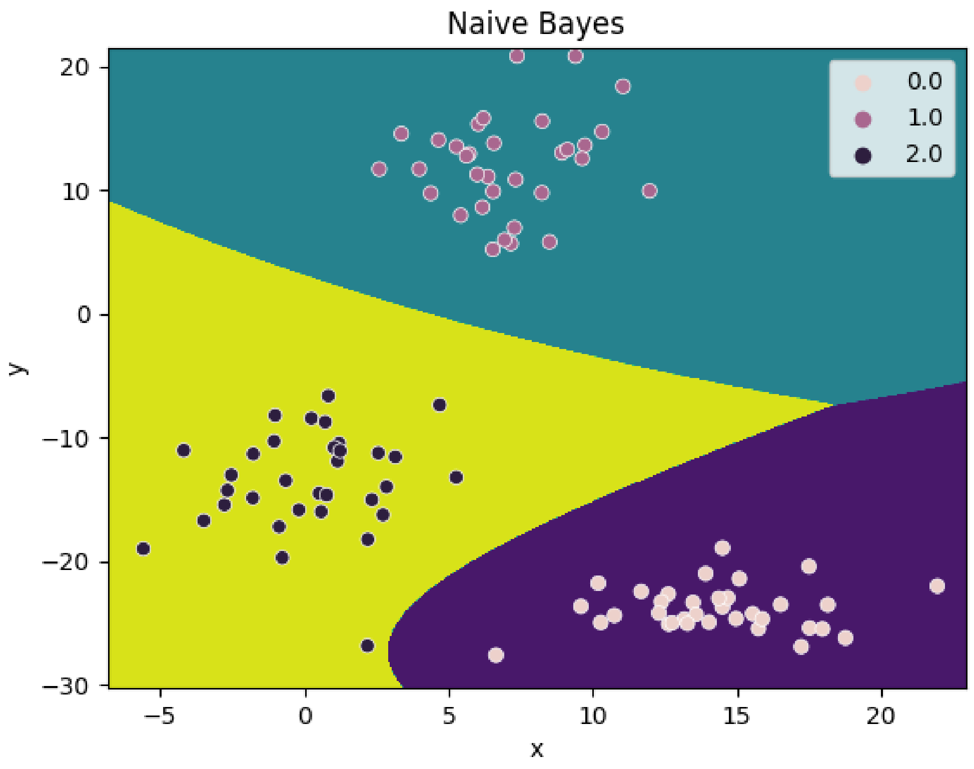
(e) (1 mark) Implement Naive Bayes classifier with covariance different for all classes on dataset2.

Solution:

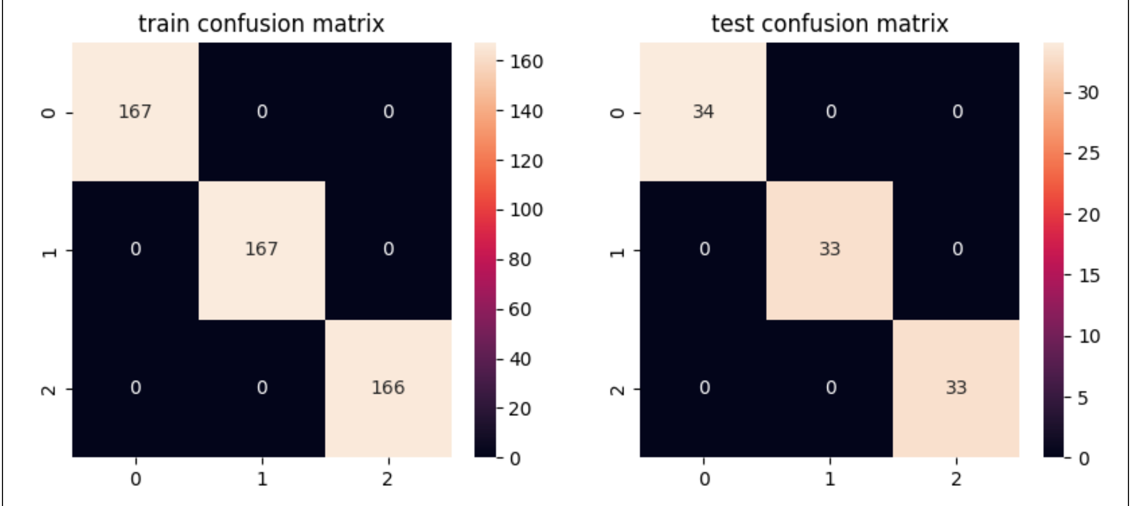
In this Question we will have to calculate the covariance matrices for each class after splitting the data given, into different classes, and use the respective covariance matrix we calculated after splitting to build the Gaussian Distribution for each class.

Accuracy for this model : Test Accuracy = 100 Train Accuracy = 100

Plot for this model:



Confusion Matrix:

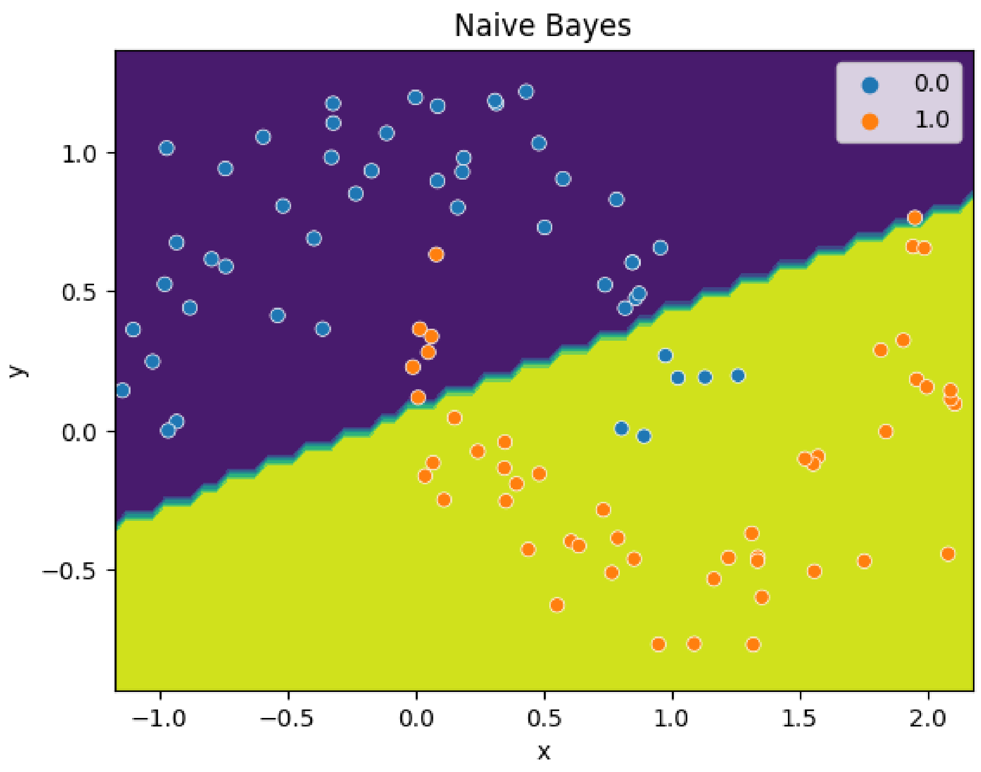


(f) (1 mark) Implement Naive Bayes classifier with covariance different for all classes on dataset3.

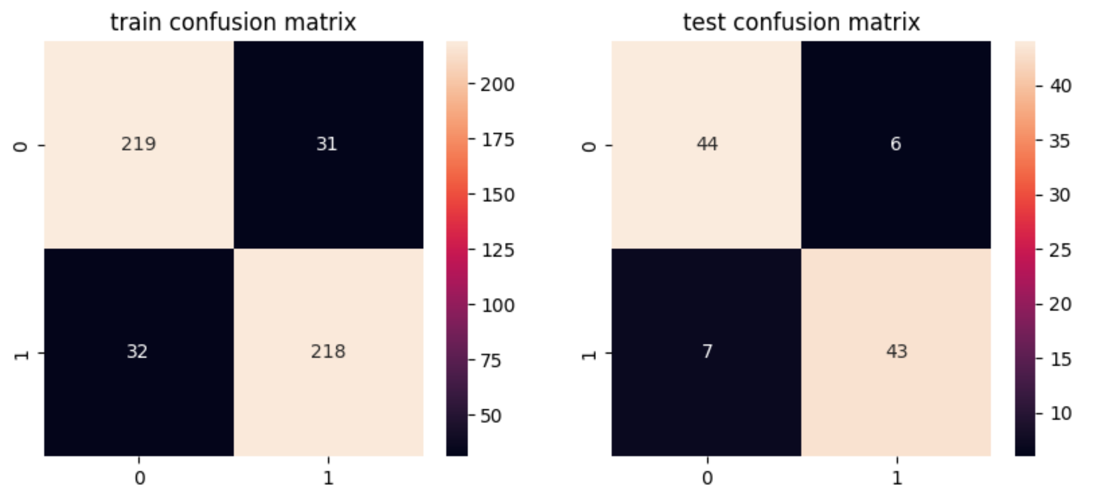
Solution:

The covariance is calculated similar to previous part - (e)
Accuracy for this model : Test Accuracy = 87 Train Accuracy = 87.4

Plot for this model:



Confusion Matrix:



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.

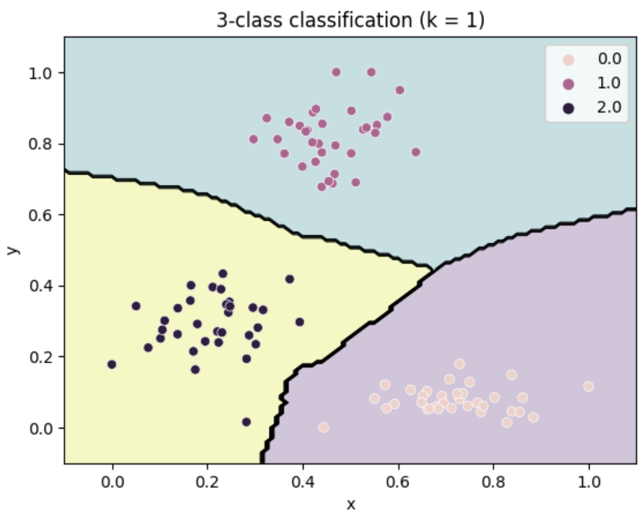
(a) (2 marks) Implement k-nearest neighbors classifier on dataset2.

Solution:

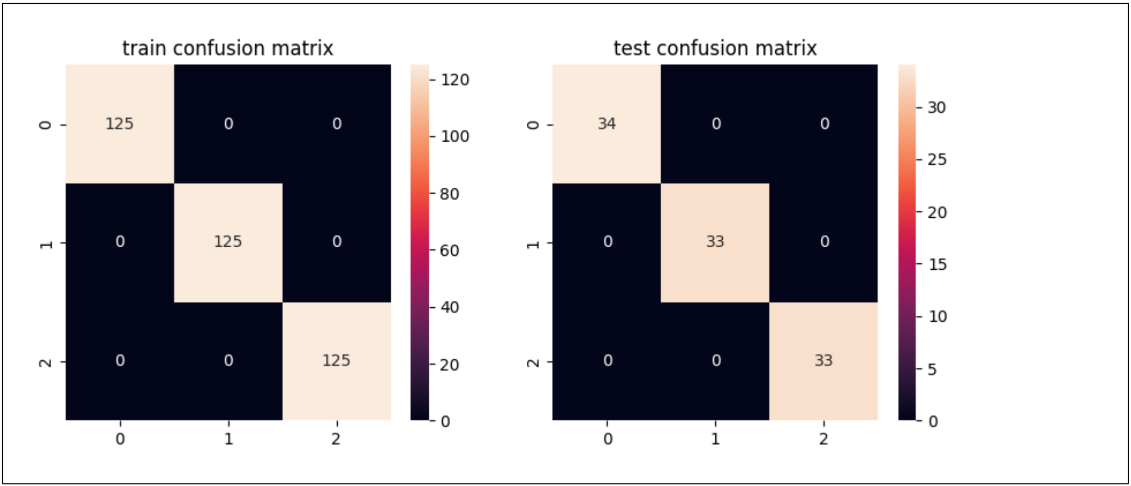
- Train set - 0.75, Validation = 0.25
- Normalization - Min-Max Scaling
- Analysis is done on 20 values of k.
- The best model is taken based on accuracy on validation data
- The best model is observed at **k=1**

Accuracy for best model: Test Accuracy=1, Train Accuracy=1

Plot for the best model:



Confusion Matrix:



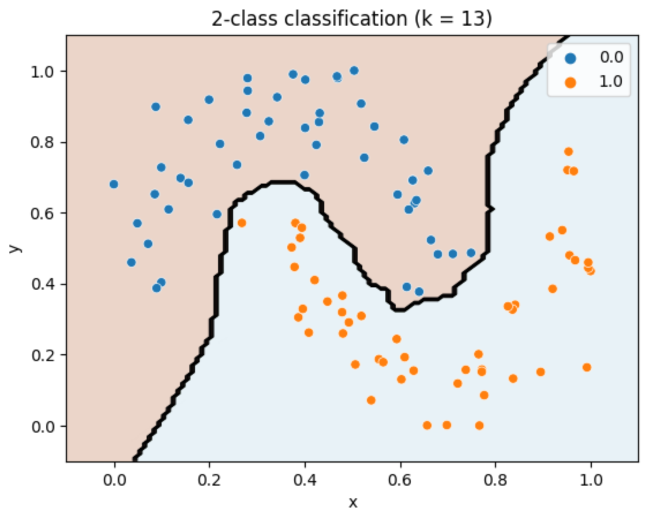
(b) (2 marks) Implement k-nearest neighbors classifier on dataset3.

Solution:

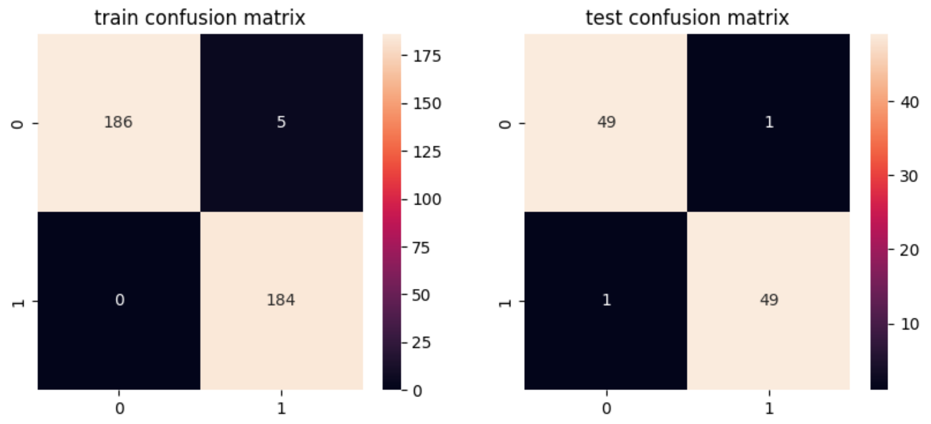
- Train set - 0.75, Validation = 0.25
- Normalization - Min-Max Scaling
- Analysis is done on 20 values of K.
- The best model is taken based on the accuracy on validation data
- The best model is observed at **k=13**

Accuracy for best model: Test Accuracy=0.987, Train Accuracy=0.98

Plot for the best model:



Confusion Matrix:


train confusion matrix
test confusion matrix