CS5691: Pattern Recognition and Machine Learning
Assignment #1

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

Teammate 1: (Debojyoti Mazumdar) **Roll number:** EE20B030
Teammate 2: (Sneha Reddy Palreddy) **Roll number:** EE20B129

- 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.

Contributions:

Debojyoti Mazumdar(EE20B030): Question-2 and Question-3-b
Sneha Reddy Palreddy(EE20B129): Question-1 and Question-3-a

Questions:

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.

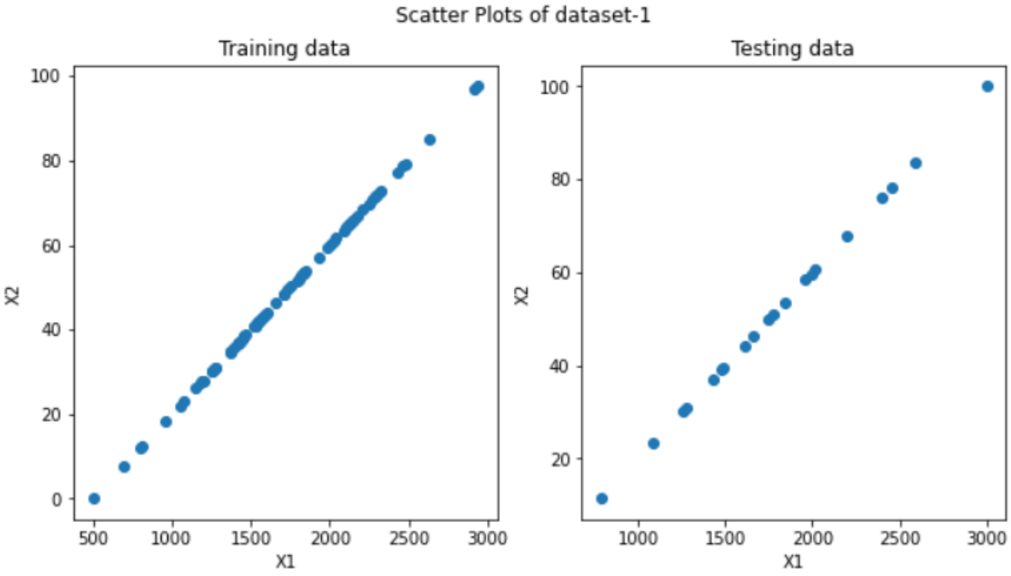
   (a) (2 marks) Use standard linear regression to get the best-fit curve. Split the data into train and validation sets and try to fit the model using a degree 1 polynomial then vary the degree term of the polynomial to arrive at an optimal solution.

   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.

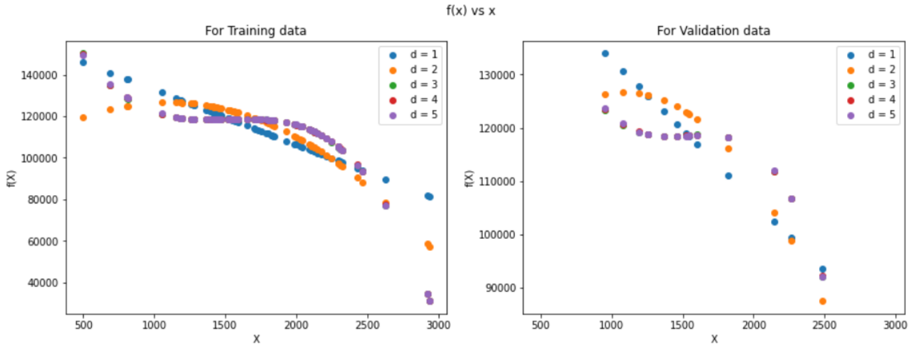
   Solution:

The figure below is a scatter plot of X1 and X2 for training and testing data. From the figure we see that X1 and X2 are perfectly correlated. So we can ignore one of the variables. So here we have ignored X2 and have done all calculations using X1 and Y.

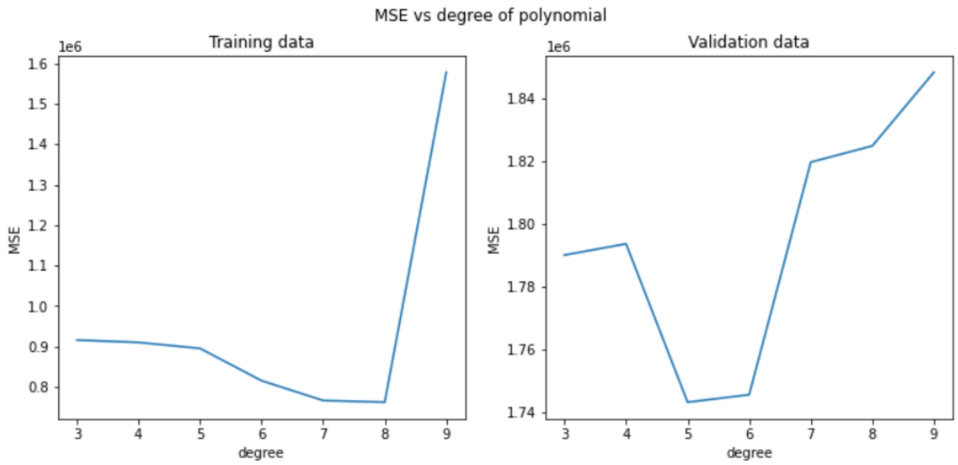


We have seperated the given training data into training and validation data in 80:20 format.

- Plot of curve of the obtained function on data points for various degree term of the polynomial for training and validation data.



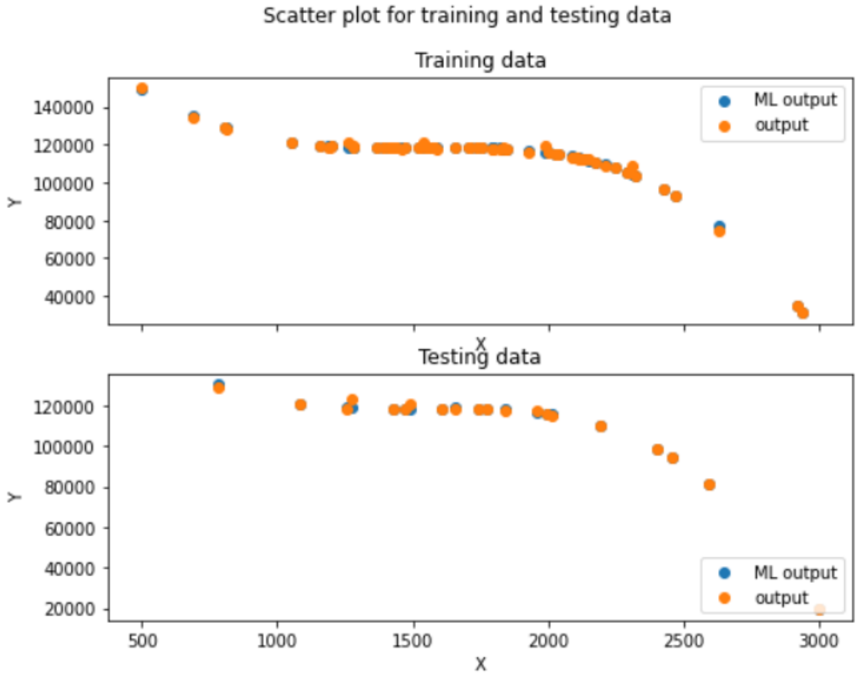
- Plot of the curve for Mean Square Error(MSE) Vs degree of the polynomial for train and validation data.



From the above MSE vs degree plot for the validation data, we get the best model would be with degree term of the polynomial equal to 5 as it has the lowest MSE.

- The error for the best model in terms of the MSE:
  MSE for training data = 894991.0458308207
  MSE for testing data = 1206154.1328425892

- Scatter plot of the best model output vs expected output for both train and test data provided:



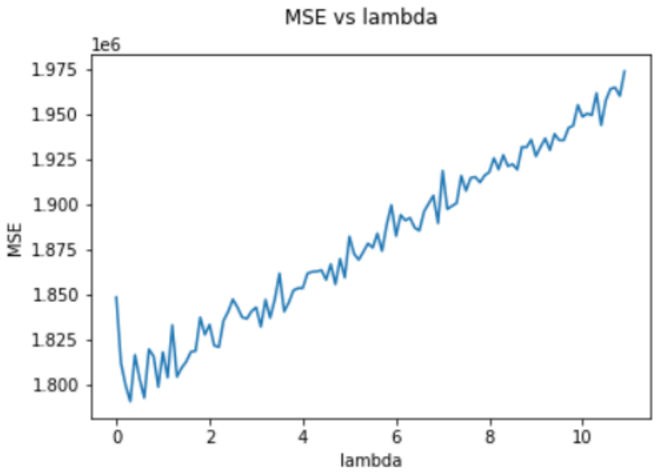
- Observations from the above plots :
  - We observe that the scatter plot of X1 and X2 is a straight line. This shows that X1 and X2 are perfectly correlated (Cov(X1,X2) = 1).
  - We observe from the scatter plot of X1 and f(X1) that as the degree increases they become more and more non-linear.
  - We observe from the plot of MSE vs degree for training and validation data that the MSE has a minimum. This is due to the "Bias-Variance trade-off" where as the degree increases the model "overfits" the training data causing high variance which leads to high MSE and for lower values of degree the model "underfits" leading to high bias which leads to high MSE.
  - We observe from MSE vs degree for training and validation data that the MSE is lower for training data than validation data.

(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.

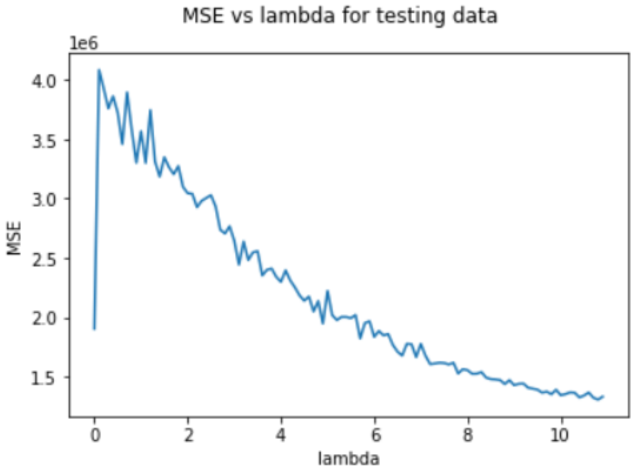
Solution:
From the plot of MSE vs degree for validation data in (a) part we see that degree = 9 overfits the data. So we will be using this model and try to control it using Ridge regression.

- Finding the MSE of the validation data using the degree 9 polynomial model and plotting it for different λ values. (Note : Here we have taken λ values with steps of 0.1). We have plotted it till $\lambda = 10$ because the MSE keeps increasing even after that.



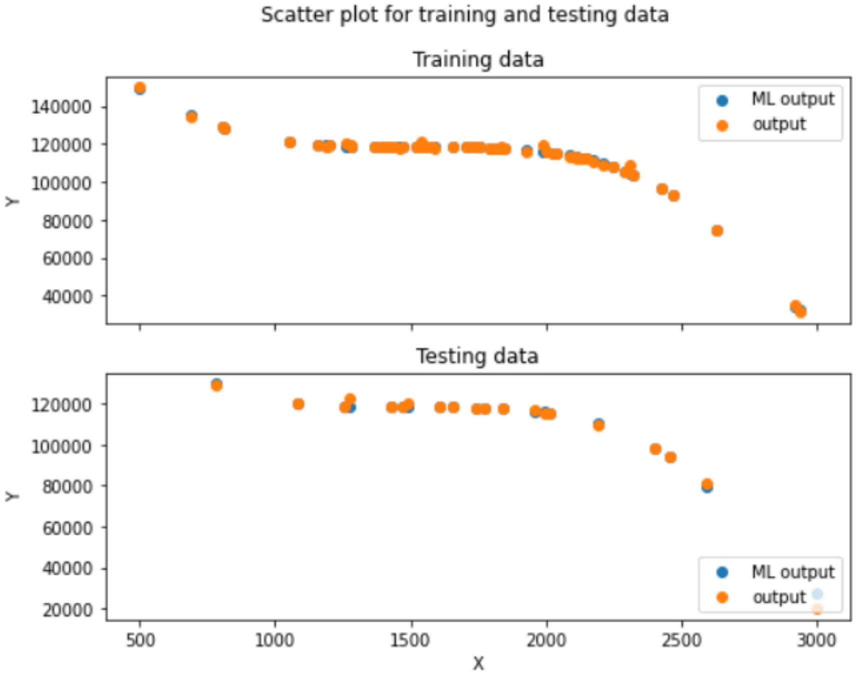
From this plot we find that $\lambda = 0.3$ gives the best model for degree 9 polynomial.

- Plot of the MSE test vs λ.



- MSE for train data = 771385.3097971738
  MSE for testing data = 3761247.415932457.

- Scatter plot of best model output vs expected output for both train and test data provided.



- Observations from the above plots:
  - We observe from the plots of MSE vs λ for training and validation data that MSE(λ) of training data is inversely related to MSE(λ) of validation data.
  - We observe that the MSE of training data is greater than the MSE for testing data for $\lambda = 0.3$.

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:

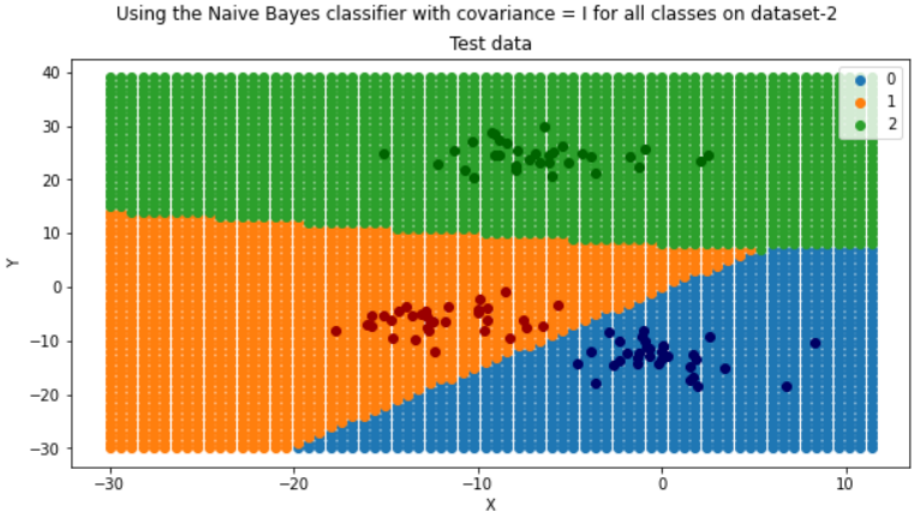
   - Accuracy on both train and test data.
   - Plot of the test data along with your classification boundary.
   - confusion matrices on both train and test data.

   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:

- Accuracy on training data is 0.996 and on the testing data is 1
- Plot of the test data along with the classification boundary.



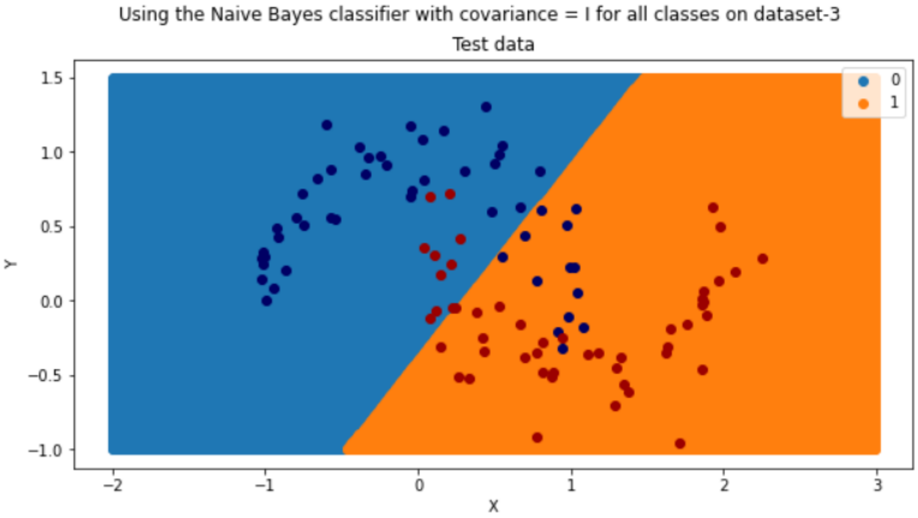
- confusion matrices on test data is :
  $$\begin{bmatrix} 34 & 0 & 0 \\ 0 & 33 & 0 \\ 0 & 0 & 33 \end{bmatrix}$$
  confusion matrix on train data is :
  $$\begin{bmatrix} 166 & 1 & 0 \\ 1 & 166 & 0 \\ 0 & 0 & 166 \end{bmatrix}$$

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

Solution:

- Accuracy on training data is 0.788 and on the testing data is 0.76

- Plot of the test data along with the classification boundary.



- confusion matrices on test data is :
  $\begin{bmatrix} 37 & 11 \\ 13 & 39 \end{bmatrix}$

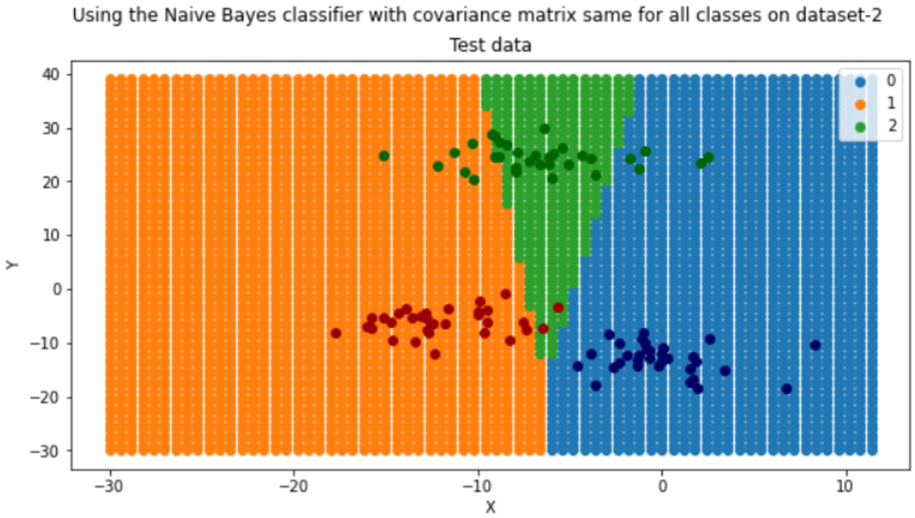
  confusion matrix on train data is :
  $\begin{bmatrix} 198 & 54 \\ 52 & 166 \end{bmatrix}$

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

Solution:

- Accuracy on training data is 0.862 and on the testing data is 0.86

- Plot of the test data along with the classification boundary.



- confusion matrices on test data is :

$$\begin{bmatrix} 34 & 0 & 5 \\ 0 & 31 & 7 \\ 0 & 2 & 21 \end{bmatrix}$$

confusion matrix on train data is :

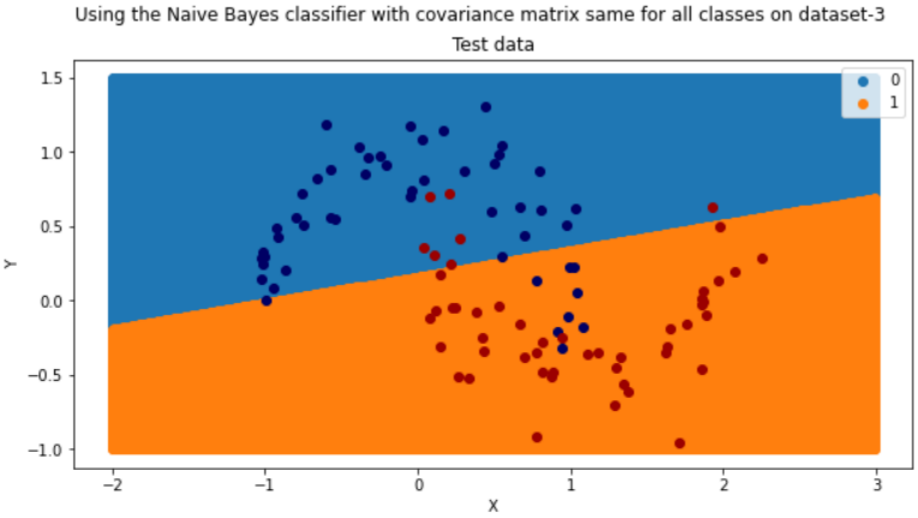
$$\begin{bmatrix} 166 & 3 & 26 \\ 1 & 159 & 34 \\ 0 & 5 & 106 \end{bmatrix}$$

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

Solution:

- Accuracy on training data is 0.856 and on the testing data is 0.85

- Plot of the test data along with the classification boundary.



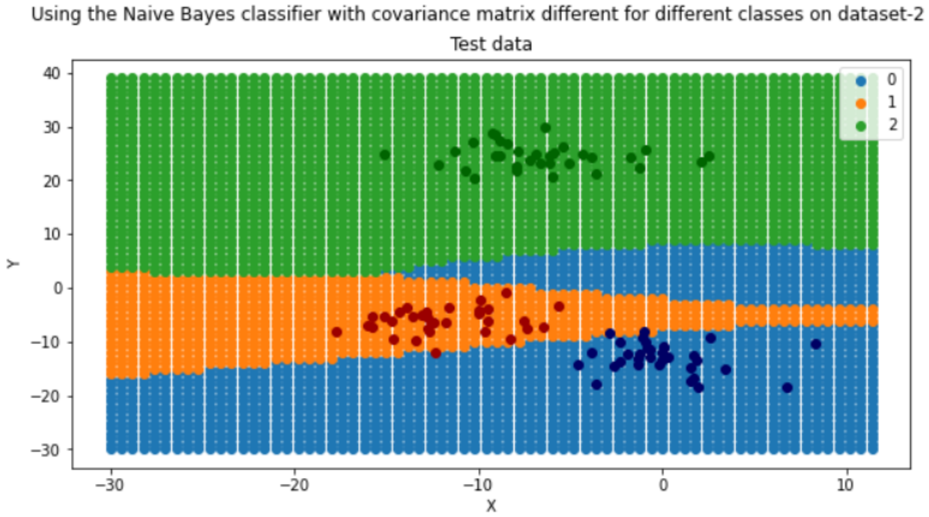
- confusion matrices on test data is :
$$\begin{bmatrix} 42 & 7 \\ 8 & 43 \end{bmatrix}$$
confusion matrix on train data is :
$$\begin{bmatrix} 215 & 37 \\ 35 & 213 \end{bmatrix}$$

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

Solution:

- Accuracy on training data is 0.986 and on the testing data is 0.97

- Plot of the test data along with the classification boundary.



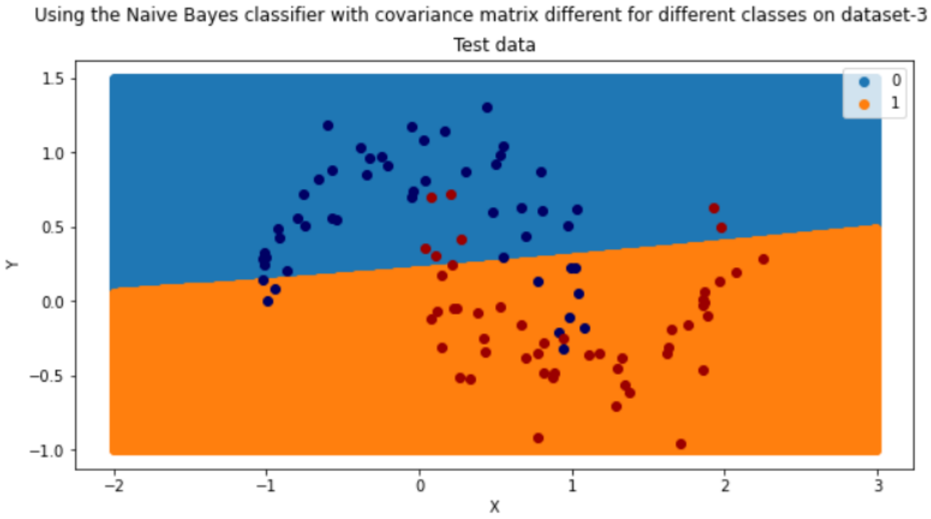
- confusion matrices on test data is :
  $$\begin{bmatrix} 32 & 1 & 0 \\ 2 & 32 & 0 \\ 0 & 0 & 33 \end{bmatrix}$$
  confusion matrix on train data is :
  $$\begin{bmatrix} 160 & 0 & 0 \\ 7 & 167 & 0 \\ 0 & 0 & 166 \end{bmatrix}$$

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

Solution:

- Accuracy on training data is 0.844 and on the testing data is 0.82

- Plot of the test data along with the classification boundary.



- confusion matrices on test data is :
$$\begin{bmatrix} 40 & 8 \\ 10 & 42 \end{bmatrix}$$
confusion matrix on train data is :
$$\begin{bmatrix} 212 & 40 \\ 38 & 210 \end{bmatrix}$$

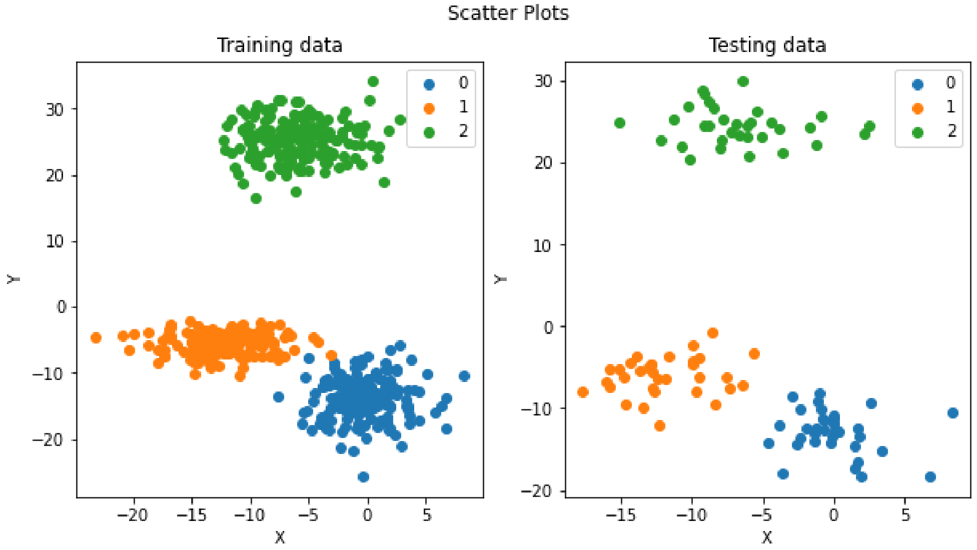
3. **[KNN Classifier]** In this Question, you are supposed to build the k-nearest neighbors classifiers on the datasets assigned to your team. Dataset for each team can be found here. For each sub-question below, the report should include the following:

   - Analysis of classifier with different values of k (number of neighbors). Split the data into train and validation sets and use validation set for finding optimal value of k.
   - 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.

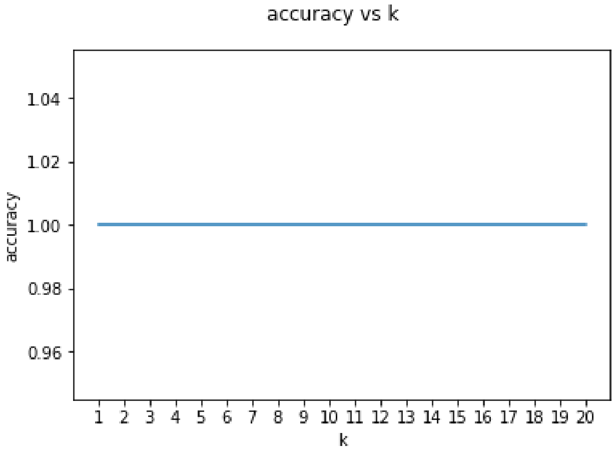
   Solution:

The figure below is a scatter plot of the training and testing data.



We split the data into train and validation data in 80:20 format.

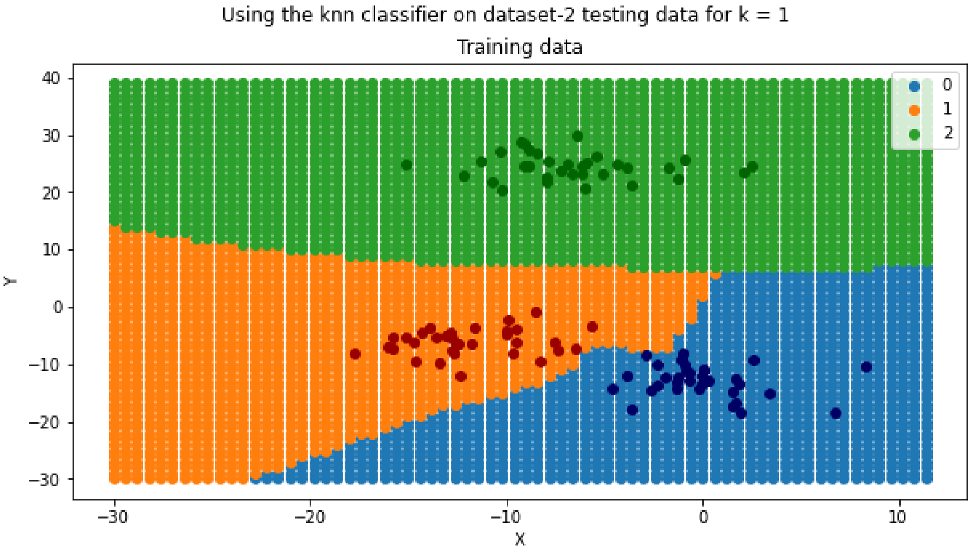
- Plot of the accuracy vs different values of K.



  As we see from the above plot, the accuracy is equal to 1 for $k \geq 1$. So we choose k=1 as our optimal value of k

- For the k=1 KNN classifier:
  accuracy on train data = 1
  accuracy on test data = 1

- Plot of the test data along with the classification boundary for the best model.



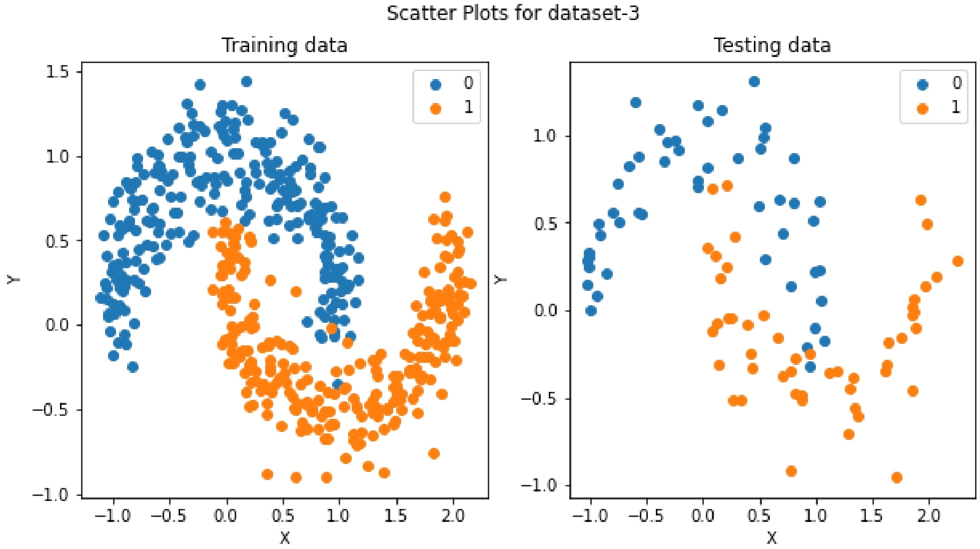
- Confusion matrix on test data : $\begin{bmatrix} 34 & 0 & 0 \\ 0 & 33 & 0 \\ 0 & 0 & 33 \end{bmatrix}$

  Confusion matrix on train data : $\begin{bmatrix} 127 & 0 & 0 \\ 0 & 135 & 0 \\ 0 & 0 & 138 \end{bmatrix}$

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

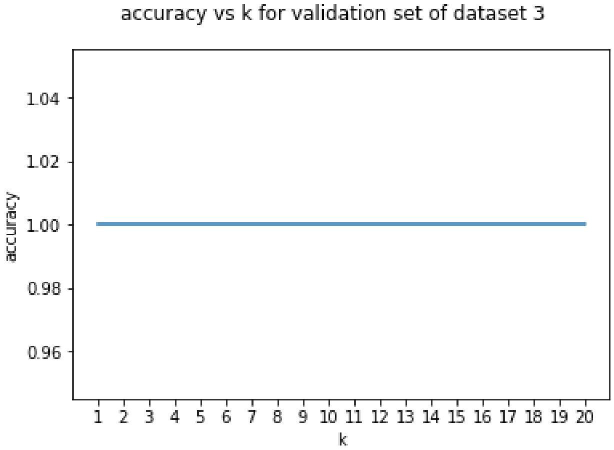
Solution:

The figure below is a scatter plot of the training and testing data.



We split the data into train and validation data in 80:20 format.

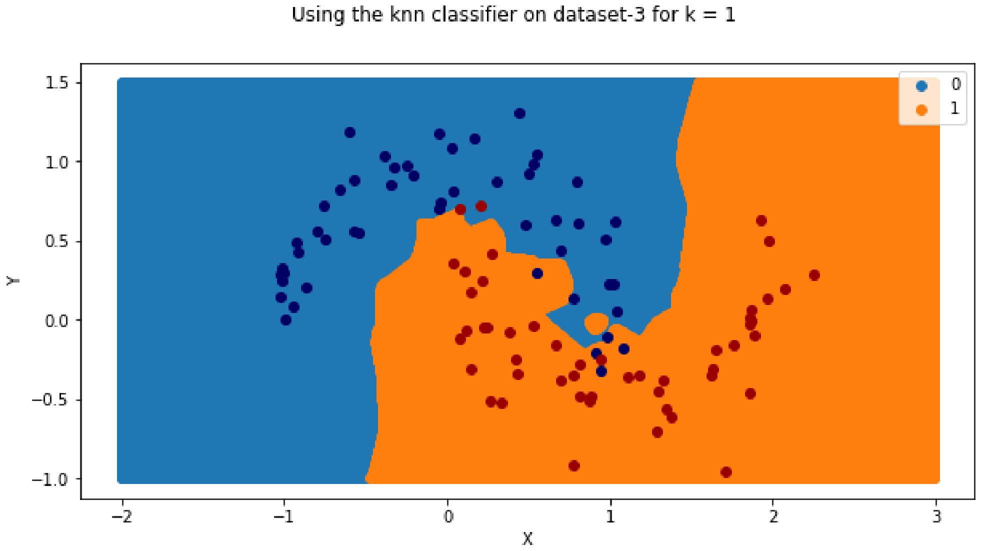
- Plot of the accuracy vs different values of K.



As we see from the above plot, the accuracy is equal to 1 for $k \geq 1$. So we choose k=1 as our optimal value of k

- For the k=1 KNN classifier:
  accuracy on train data = 1
  accuracy on test data = 0.95

- Plot of the test data along with the classification boundary for the best model.



- Confusion matrix on test data : $\begin{bmatrix} 47 & 2 \\ 3 & 48 \end{bmatrix}$

  Confusion matrix on train data : $\begin{bmatrix} 198 & 0 \\ 0 & 202 \end{bmatrix}$