CS5691 Assignment 2

Amogh Patil (EE19B134), Rishabh Adiga (EE19B135)

March 2022

1 Regression

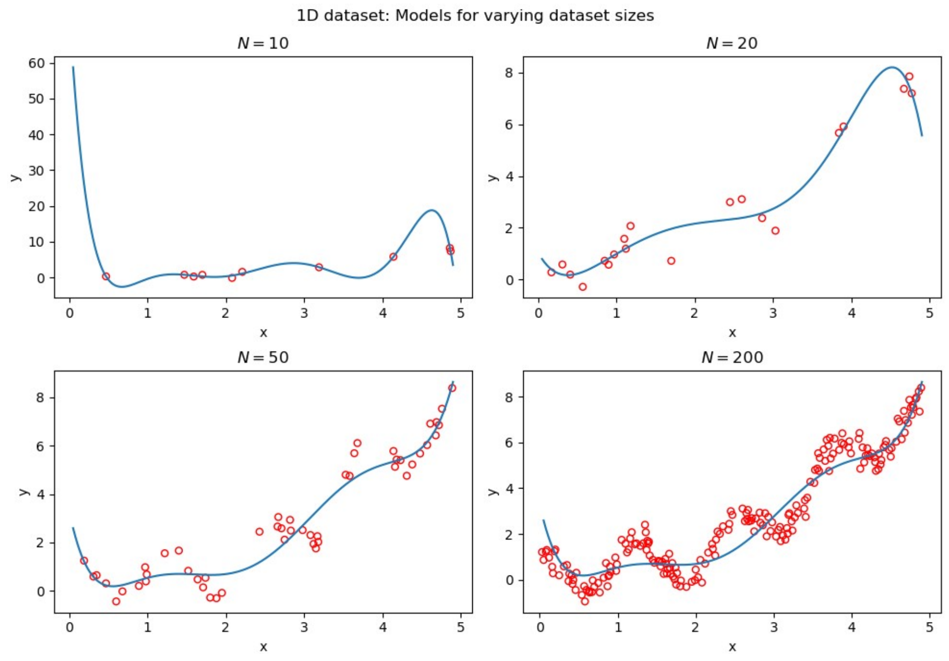
1.1 Introduction

Given 2 datasets with 200 samples of 1 dimensional data and 1000 samples of 2 dimensional data, our aim was to build a good regression model on each of these datasets and also observe the effects on the model when various factors like dataset size, model complexity,etc are varied.Various plots and inferences were made based on these experiments.All models were made using the normal equation on the given data with a polynomial function.

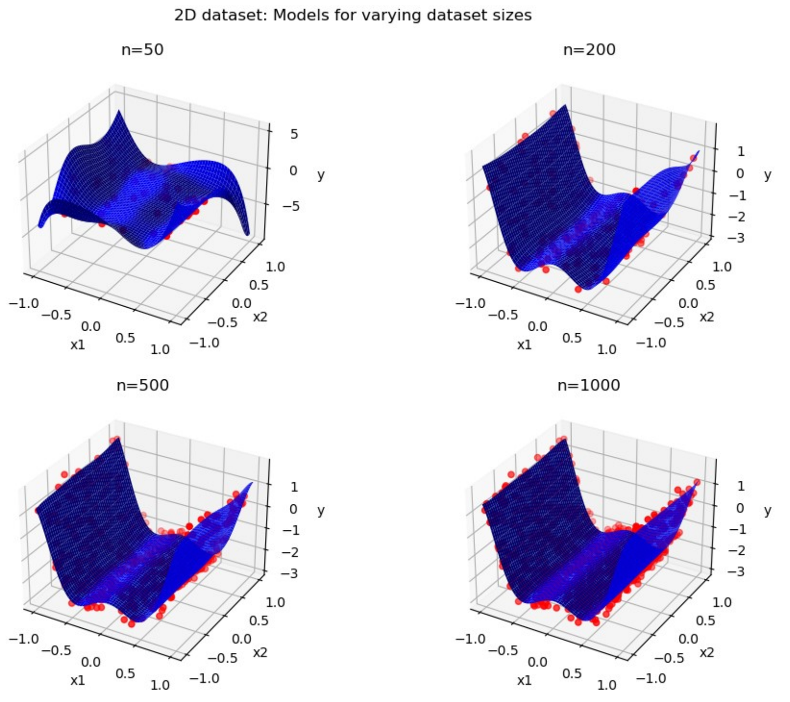
1.2 Experiments and Inferences

1.2.1 Regression Model obtained for different dataset sizes

The provided training data is uniformly sampled in datasets of different sizes.For the 1D data, dataset of sizes 10,20,50,200 were created and models for these were constructed.For the 2D data,dataset of sizes 50,200,500,1000 were created and models for these were constructed.
The following plots were obtained for the models formed using 1D data:



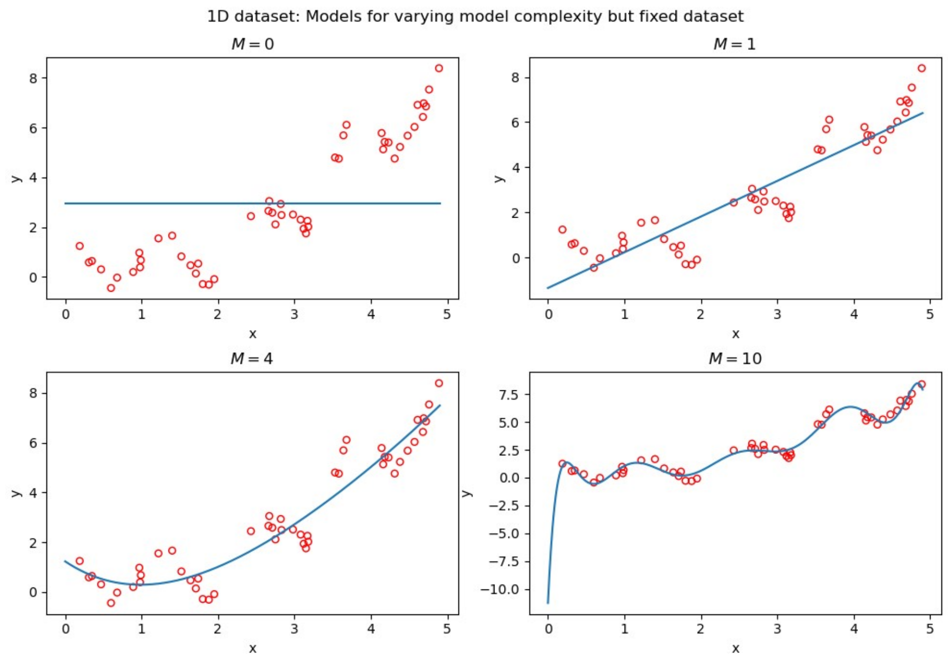
The following plots were obtained for the models formed using 2D data:



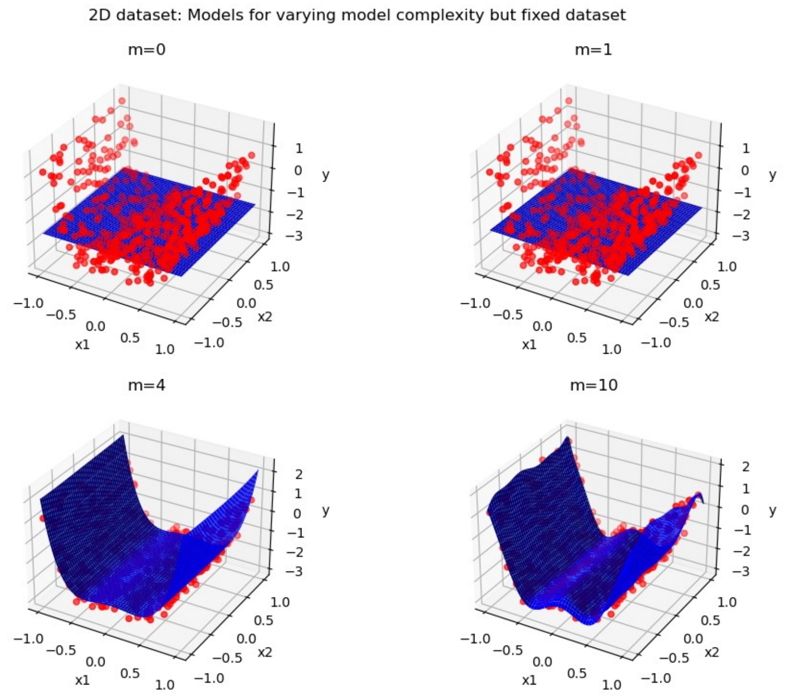
We can see and infer that, for a given model complexity,the over-fitting problem become less severe as the size of the data set increases.Another way to say this is that the larger the data set, the more flexible the model that we can afford to fit to the data.A rough heuristic that can be used is that the number of data points should not be less than 5 or 10 times the number of parameters in the model.

1.2.2 Regression Model obtained for different model complexities

In this experiment, we take a fixed set of randomly sampled data points and observe the models for different model complexities(Polynomial order).For the 1D and 2D data, models with polynomial orders of 0,1,4,10 were formed. The following plots were obtained for the models formed using 1D data:



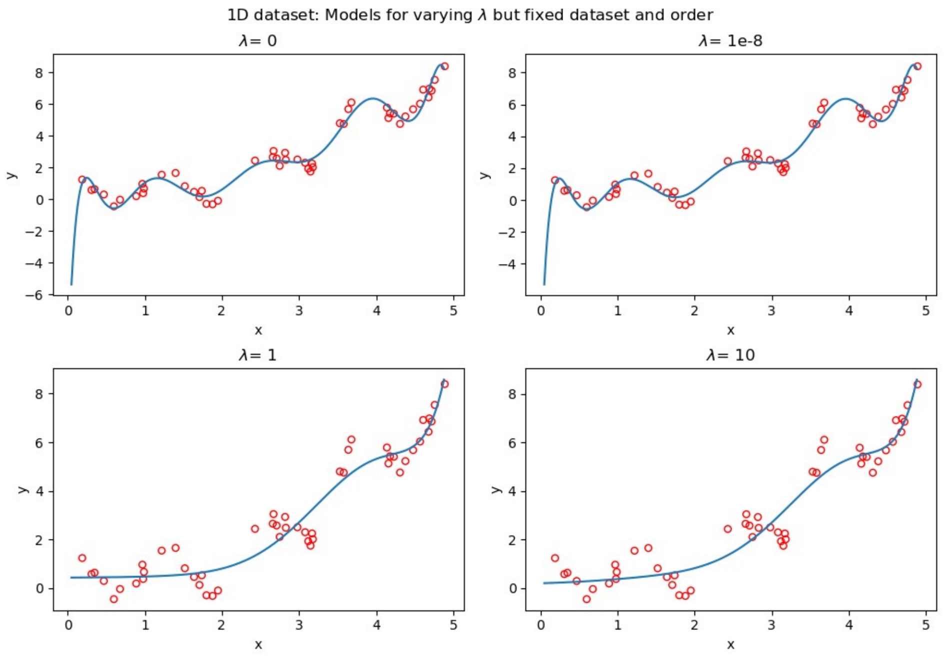
The following plots were obtained for the models formed using 2D data:



We see that, as M increases, the magnitude of the weights gets larger.The weights have become finely tuned to the data by developing large positive and negative values so that the corresponding polynomial function matches each of the data points exactly, but between data points (particularly near the ends of the range) the function exhibits the large variation and changes.Intuitively, what is happening is that the more flexible polynomials with larger values of M are becoming increasingly tuned to the random noise on the target values and hence larger M results in over-fitting of the data.

1.2.3 Regression Model obtained for different regularisation parameter values

In this experiment, we take a fixed set of randomly sampled data points and a fixed polynomial order of m=10 and observe the models for different regularisation parameters(λ).We use λ=0,1e-8,1,10 for the different cases.The plots obtained are as follows:



We see that, for a value of $\lambda = 1$, the over-fitting has been suppressed and now the model has started to under-fit.Increasing λ too much will lead to a poor fit too and hence, there is an optimal value of λ that can be found.Increasing λ results in the reduction in the magnitudes of the weights of the polynomial.

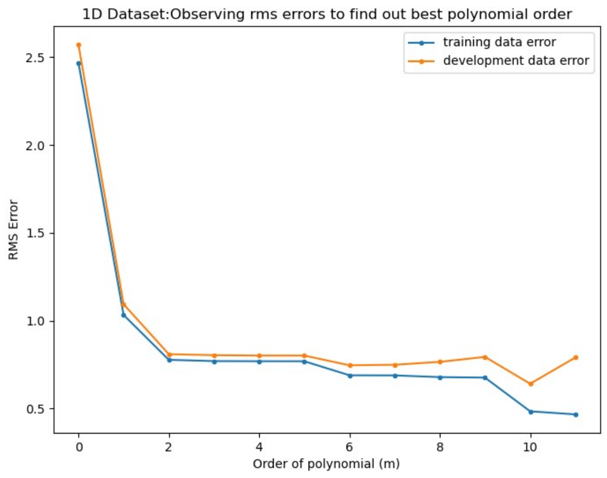
1.3 Finding Best Models

To obtain the best fit, we make use of the development data provided to optimize our hyper-parameters.
Hyperparameters that are optimized are:

- Order of polynomial (m)
- Regularisation parameter (λ)

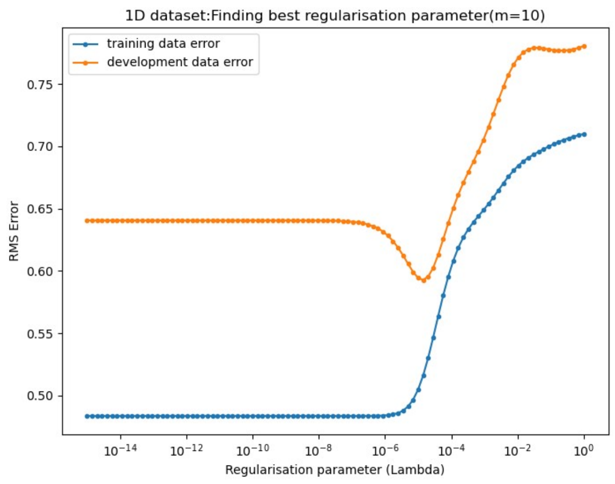
1.3.1 1D dataset best model

First we plot the graph of the root-mean-square error, evaluated on the training data and on the development data for various values of m(order of polynomial).The plot obtained is as follows:

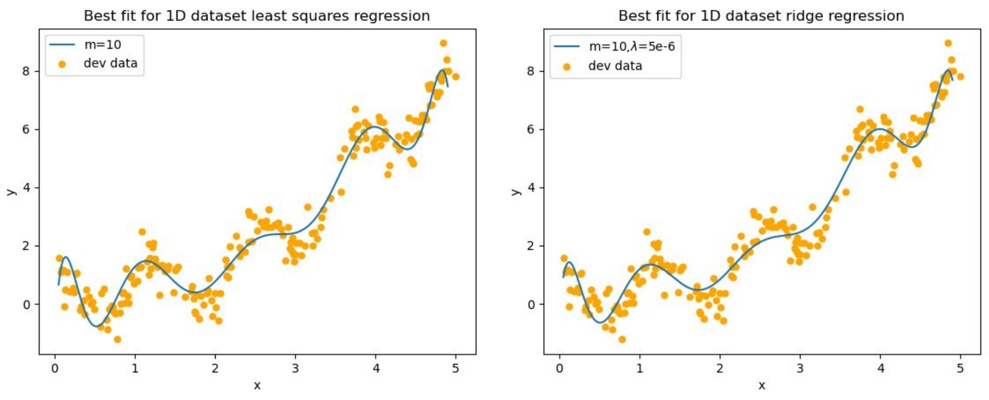


From the above graph, we can see that after m=11,the training error reduces, but the error on development data starts to increase and this is a sign that our model will not scale well on new unseen data for polynomials of order greater than or equal to m=11.Therefore the best polynomial order to pick is m=10.

Next, we plot the graph of the root-mean-square error versus Regularisation parameter (λ) for the m = 10. polynomial.The plot obtained is as follows:

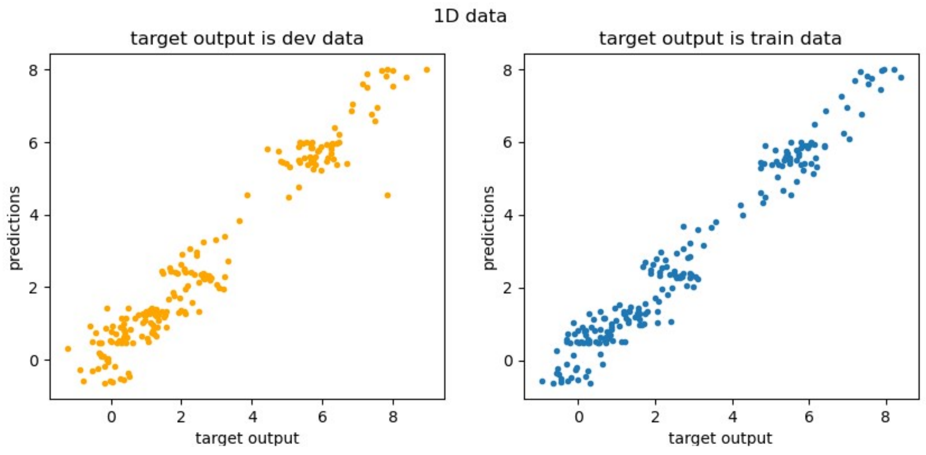


We see a drop in the development data error at around λ=5e-6.This means that our model is scaling better to new unseen data for regularisation parameters in this region of the graph.So the best value for λ=5e-6.
The best models are shown below along with the rms errors:



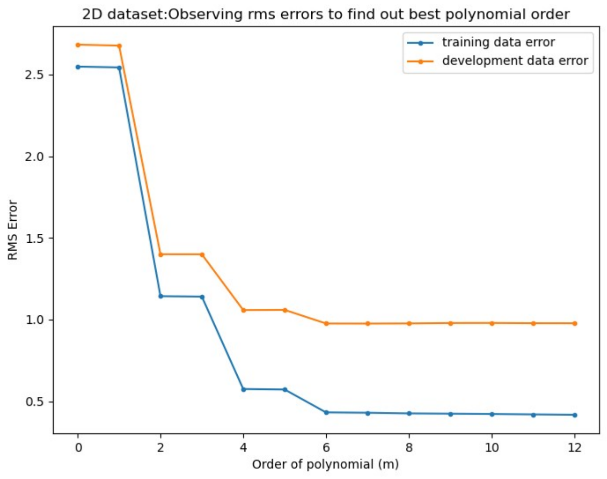
- RMS error for best model on 1D dev data for **Least square regression** = 0.640616875
- RMS error for best model on 1D dev data for **ridge regression** = 0.605578278

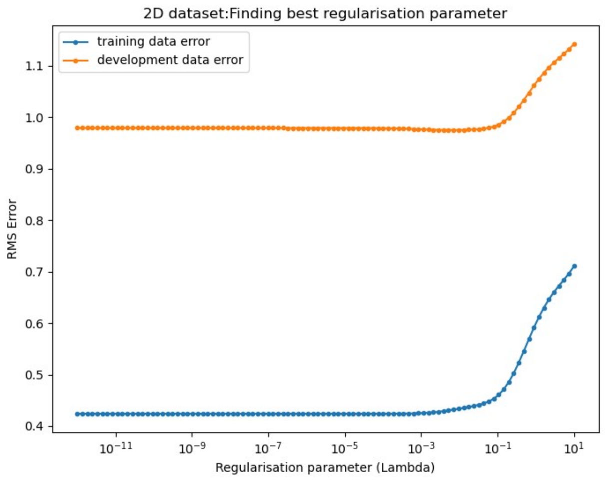
The Scatter plot with target output on the x-axis and model output on the y-axis for the best performing model is shown below(as expected, it approximately forms y=x line):



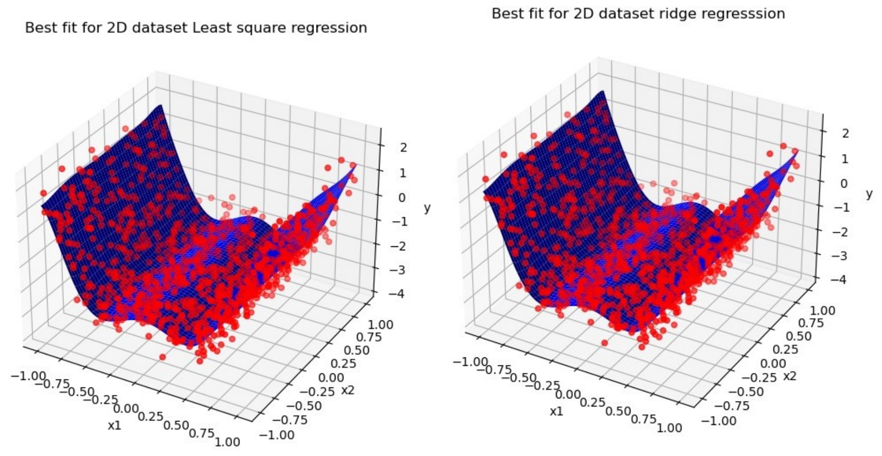
1.3.2 2D dataset best model

The same method is followed for 2D data too.The plots obtained are as shown below:



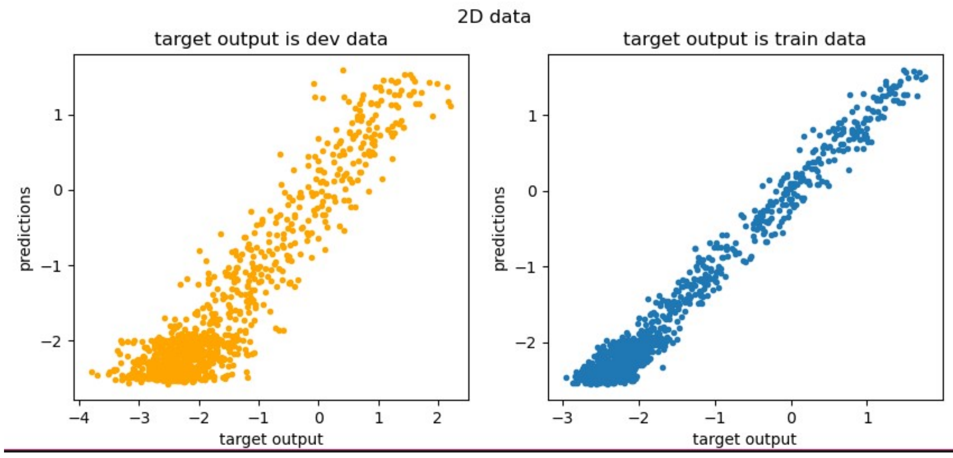


Therefore, the order of the polynomial chosen is m=9 as the error has become constant and the regularisation parameter is λ=1e-5.
The best models are shown below along with the rms errors:



- RMS error for best model on 2D dev data for **Least square regression** = 0.9792
- RMS error for best model on 2D dev data for **Ridge regression** = 0.9790

The Scatter plot with target output on the x-axis and model output on the y-axis for the best performing model is shown below(as expected, it approximately forms y=x line):



2 Bayesian Classifier

2.1 Introduction

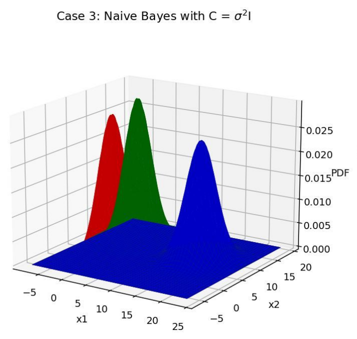
Given 3 datasets of 2 dimensional data, our aim was to build a good Bayesian Classifier model on each of these datasets and also find good approximations for different forms of the covariance matrix and visualize the effects. Various plots and inferences were made based on these experiments.

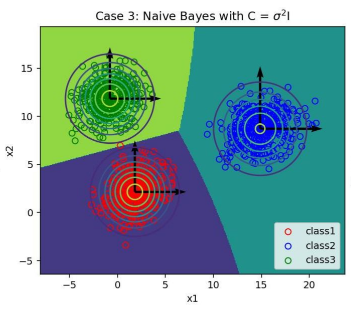
2.2 Experiments and Inferences

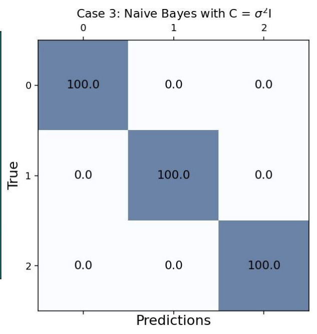
Note-We have ran all cases for all 3 datasets but are showing only a few in the report.The ones that are shown here, are the cases performing the best for the respective datasets.

2.2.1 Linearly Separable Data

CASE 3 - Naive Bayes with Covariance = σ^2 I



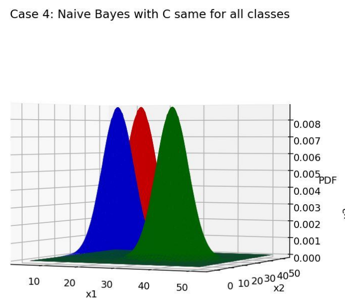


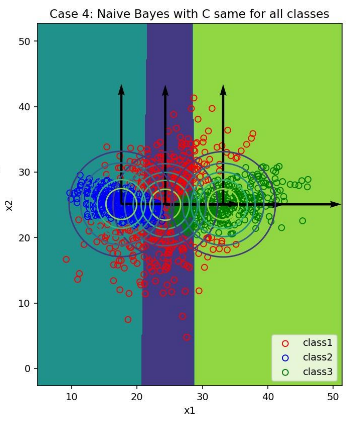


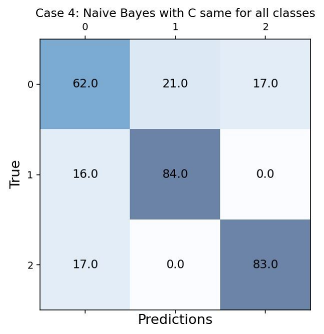
The Model classifies the Development Data perfectly(100% accuracy).The contours are circular as expected.The Decision Boundaries are Linear as expected.We can also see that the eigenvectors are parallel to x and y axis since we are using diagonal covariance matrices.

2.2.2 Non-Linearly Separable Data

CASE 4 - Naive Bayes with Covariance same for all classes



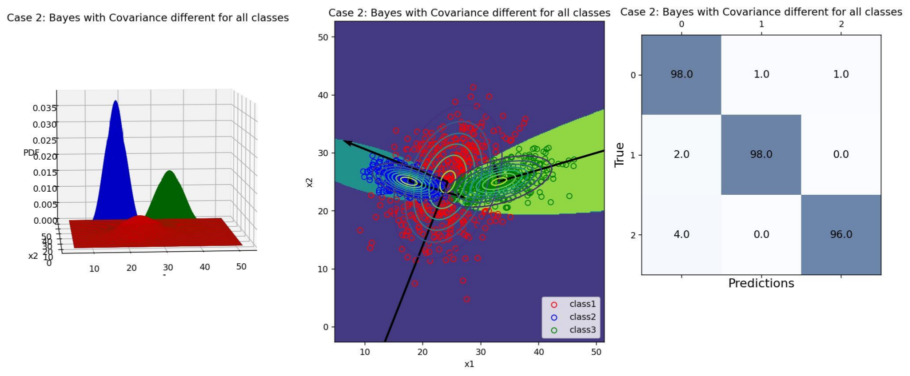




Underfitting of Data is observed since the model is not complex enough. The results on the dev data are average. The number of True Positives is low as seen from Confusion Matrix as the model is attempting to classify non-linear data with linear Decision boundaries.

```
Accuracy on Test Data : 76.33333333333333 %
```

CASE 2 - Bayes with Covariance different for all classes

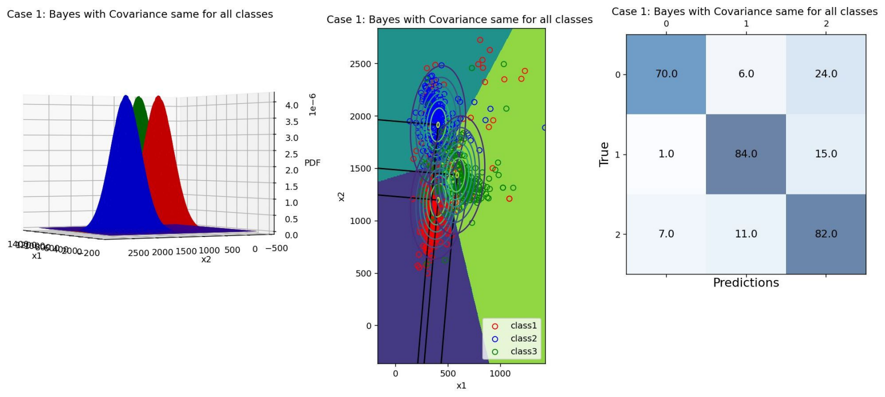


This Model classifies the Development Data with highest accuracy for this data set.We can see that it has captured the alignment and spread of the 3 classes as each covariance matrix is different(hence hyperquadratic decision boundries)

`Accuracy on Test Data : 97.33333333333334 %`

2.2.3 Real data

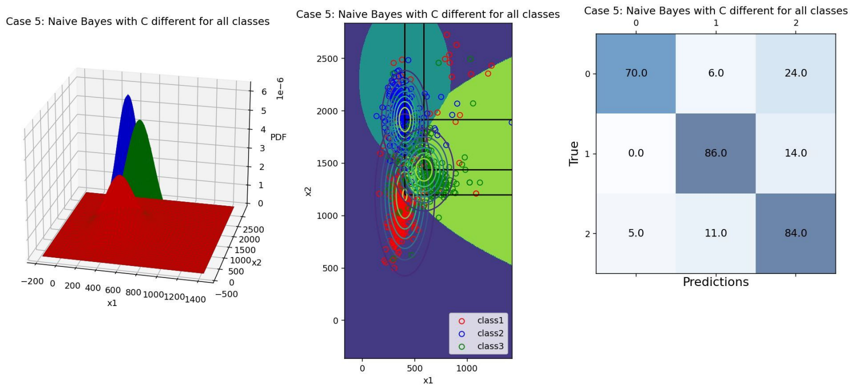
CASE 1 - Bayes with Covariance same for all classes



Underfitting of Data is observed since the model is not complex enough. Linear Decision boundaries are not capable of classifying the data set.

`Accuracy on Test Data : 78.66666666666666 %`

CASE 5 - Naive Bayes with C different for all classes



This model performs better than Case 2(Bayes with C different for all classes). Real data is more likely to be noisier, hence the Naive Bayes model with lesser model complexity reduces the amount of overfitting taking place.

`Accuracy on Test Data : 80.0 %`

2.3 Receiver Operating Characteristic and Detection Error Tradeoff curves

2.3.1 Linearly separable data

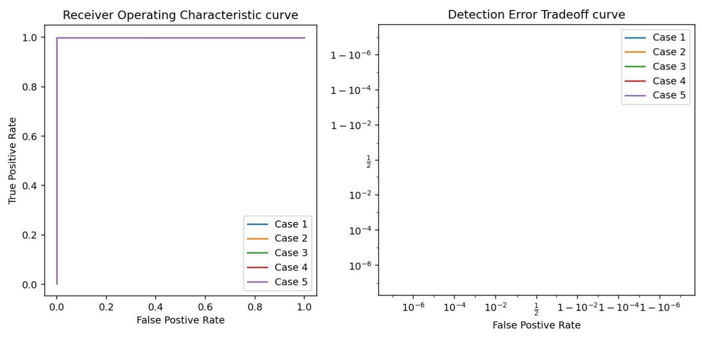


Figure 1: Linearly Seperable Data

For linearly separable data ,ROC for all cases is perfect and are overlapping as they are 100% accurate. Since the model performs really well, the FPR and FNR is nearly zero for most thresholds hence we dont see any curve on the DET plot.

2.3.2 Non Linearly separable data

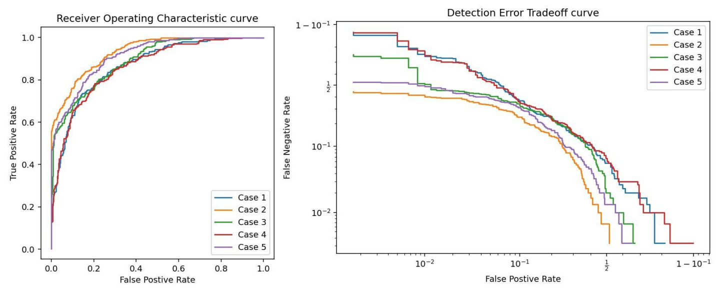


Figure 2: Non-Linearly Separable Data

Case 2 model performs the best as the area under the ROC curve is the highest and the DET curve is closest to the Origin for this case too.So Bayes classifier with Covariance different for all classes is the best for this data.

2.3.3 Real data

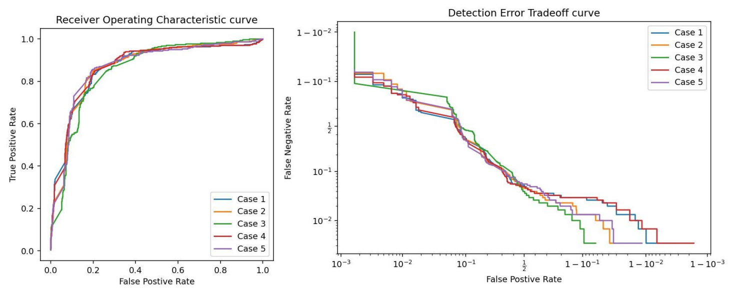


Figure 3: Real Data

Case 5 model performs the best but the differences between the cases is very small as real data is nosier and models trained on this data find it harder to scale up to new unseen data.But from the confusion matrices,ROC and DET curve, we see that Naive Bayes with C different for all classes is the best performing model.