LAB 4 REPORT

NAME: [redacted] **ROLL NO.** [redacted]

Q1

A) Pre-Processing

- For pre-processing it was seen that many rows contained NaN values. Dropping all these rows would lead to considerable data loss. On inspection of data,it was observed that most NaN values belonged to the ‘Cabin’ and ‘Age’ Column. Hence removing the ‘Cabin’ and ‘Age’ column was necessary.
- Other columns such as ‘PassengerId’, ’Name‘, ’Fare’, ’Ticket’ were unnecessary and hence were dropped , the remaining feature column ‘Sex’, ‘Pclass’, ’Embark’ seemed to be discrimintaing feature and hence they can’t be dropped. These columns being categorical were further encoded for model implementation and training.
- The data was splitted into training data and testing data in ratio(75:25).

B) Identifying best variant for Naive Bayes Classifier

- Since the remaining columns were categorical, use of Gaussian Naive Bayes classifier was to be avoided.
- For deciding between Multinomial or Bernoulli classifiers, I had checked the accuracy of both variants from inbuilt library (sklearn) , Multinomial Naive Bayes Classifier gave best accuracy and hence it was chosen for implementation.

C) Implementing Multinomial Naive Bayes Classifier

- For implementing the model, first prior probabilities of each class were calculated.
- Next the data was divided into sub data on the basis of the ‘Survived’ column.

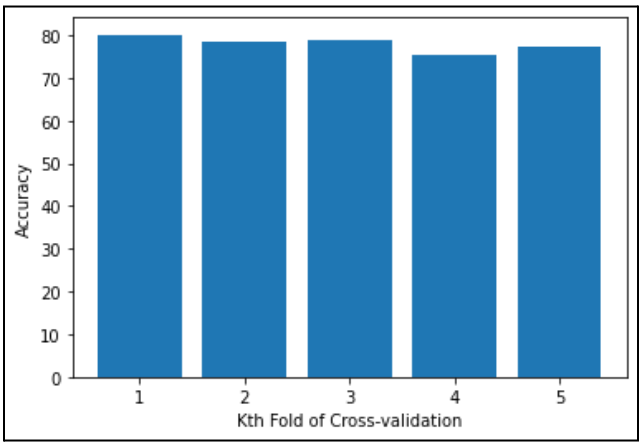
- The likelihood/ class conditional probability distribution for each feature for every sub data was calculated and stored in a dictionary.
- During prediction the model we will compare P(C/X) = (Likelihood(X/C)*(Prior(C)) for each class and the class with maximum P(C/X) will be our prediction.

D) 5 Fold Cross Validation

- For cross validation , the training data was divided into 5 parts , the classifier was trained on 4 parts and tested on the remaining part each time its accuracy was noted. The average accuracy was found between 75 to 80% .

E) Visualize Cross Validation Result and Compute the probability of the top class for each row in the testing dataset.

- For visualization cross validation result a graph was plotted between accuracy for every set.



- From the graph we can summarize that the accuracy for each fold was around 75 to 80% hence we can say our model works does not suffer overfitting or underfitting.
- For computing the probability of top class for each row in the testing set, the likelihood of data for each class calculated and multiplied by the class's prior giving us posterior of that class, the class that gave us maximum

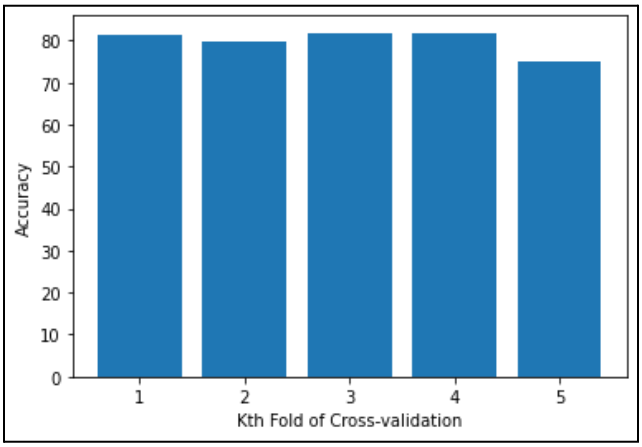
value of posterior probability was our top class and this posterior probability was stored and printed.

F) Comparing Probability with inbuilt Naive Bayes classifier

- Inbuilt Multinomial Naive Bayes classifier was implemented from sklearn and its accuracy was observed to lie around 56 to 60%, which is lesser than the accuracy of the model implemented earlier.

G) Implementing any other model , performing cross validation and comparing performance to my model

- The model I implemented in this part was Decision Tree Classifier , I implemented it from sklearn library.
- Average Accuracy of DTC was found around 78-80 %. Individual accuracy on each set of cross validation did not vary significantly and stayed around 75 to 82%.

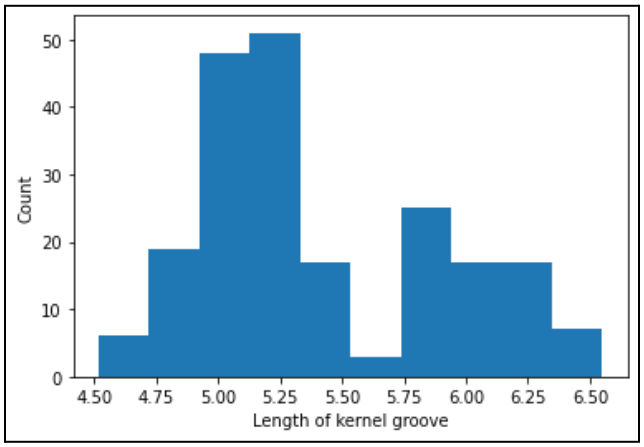


- On comparing DTC to my Multinomial NB classifier, the average accuracy in cross validation was nearly the same for both. Also the accuracy on testing data was nearly the same for both models.
- Reason for such similar accuracy can be attributed to the fact that both DTC and Multinomial NB Classifiers are made such that they perform best when working with categorical data.

Q2)

A) Plotting histogram for distribution of sample

- Histogram bar graphs were plotted for every feature in data.



(Multiple graphs are plotted but I have shown only 1 graph in report)

B) Calculating Prior Probability for each class

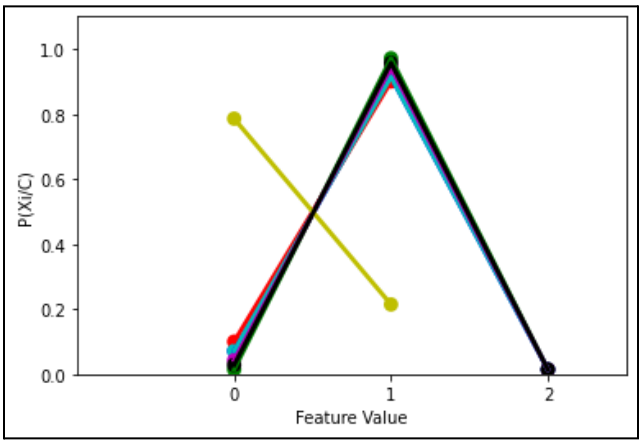
- For calculating Prior Probability for each class, the number of data points corresponding to that class was divided by the total number of data points.

C) Binning from scratch

- For implementing binning , say the number of bins is ‘n’ for each feature it was divided into n parts such that for each part difference between the maximum and minimum value were same (or nearly same) and the range of each part was disjoint from each other.
- This process was repeated for bin size from 1 to 10 and the optimal bin number was decided on the basis of entropy.
- Since entropy is a measure of impurity in data, a low entropy will mean that the bins created are homogeneous (similar values are categorized together). Hence bin size that gave minimum entropy was selected.

D) Likelihood /class conditional probabilities for each class

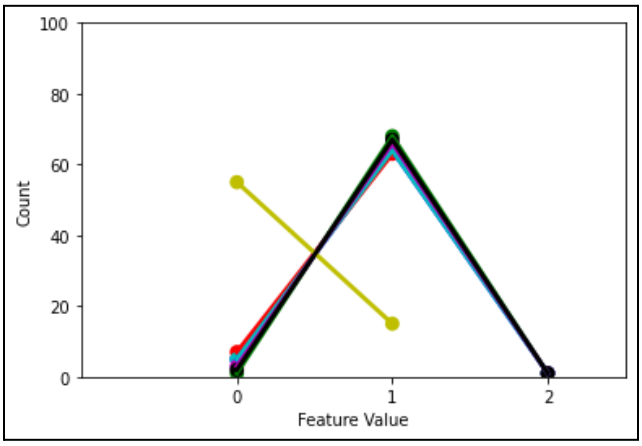
- For a class, class conditional distribution corresponding to every feature was calculated by finding the frequency for all feature values and dividing it by the total number of data points corresponding to this class.
- This process was repeated for every class and a combined graph for each class conditional probability (of each feature) was plotted.



(Multiple graphs are plotted but I have shown only 1 graph in report)

E) Count of each unique element for every class and comparing it with plot of distribution

- For every feature, the count of every feature value was calculated and stored.
- This process was repeated for every class and a combined graph of each feature was plotted.

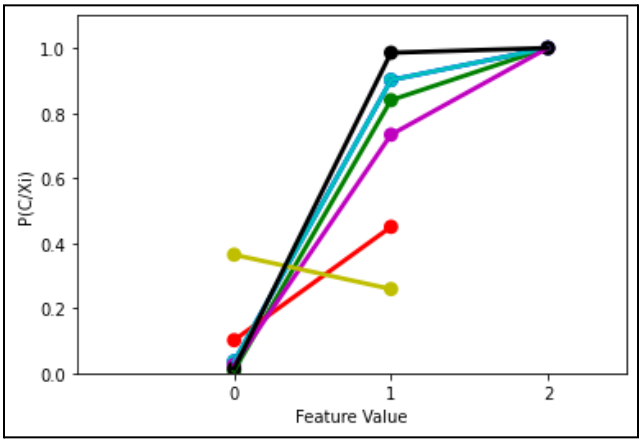


(Multiple graphs are plotted but I have shown only 1 graph in report)

- By comparing these graphs and plot of distribution we can say that for most features the count of unique value or the distribution is most dense around the mean or middle of range of features .

F) Calculating and plotting posterior probability

- To calculate the posterior probability of a particular class , the likelihood value (corresponding to this class) of the feature value was multiplied by the prior of the class and divided by the probability of the feature value. The distribution obtained was stored.
- This process was repeated for every class and graphs of posterior probability distribution of every class were plotted.



(Multiple graphs are plotted but I have shown only 1 graph in report)

- From analyzing the plot I have 2 things, the prior probability of class only does the scaling of posterior distribution from likelihood graph in y direction while probability of feature value changed the line graph.
- While comparing posterior probability of different classes, prior probability of class will be more useful (as it can vary across classes for a feature value) and probability of feature will be less useful as it will be the same across all classes.