Lab Assignment 3
Prakhar Gupta
B21AI027

Question 1:

Part 1-

- Downloaded dataset using wget command called using os.system
- Load **titanic.csv** in df using pd.read_csv
- Dropped **“Name”,"Ticket","Cabin","PassengerId"** columns using df.drop
- Checked for not filled rows using df.isnull().sum()
- Filling not filled values in “Age” column by rounded mean of whole column using fillna
- Removing two rows 61,829 as they have NaN in the “Embarked” column
- **Ordinal encoding** “Sex” column
- Plotting the data of every column using sn FacetGrid and sns.histplot and sns.countplot()
- **One hot encoded** “Embarked” column using pd.get_dummies
- Making df to X,y by y=df[“Survived”] and X=df.drop([“Survived”],axis=1)
- Using train_test_split to split the X,y

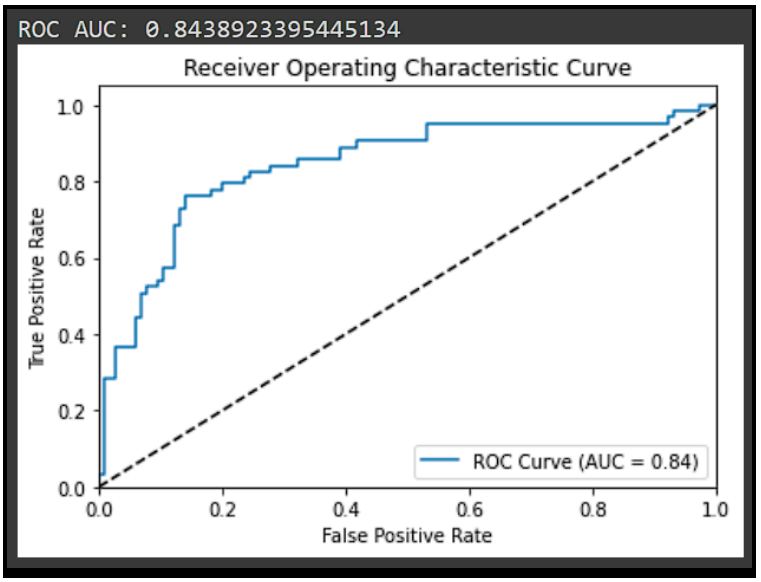
Part 2-

- Features Age and Fare are continuous and Features which are categorical are(Pclass, Sex, Embarked, and Survived).
- So best variant is gaussian naive bayes
- This classifier looks for **continuous features** which are **normally distributed** and it fulfils our condition in dataset this classifier is based on the probabilistic approach and Gaussian distribution

Part 3-

- Implemented gaussian naive bayes classifier using scikit
- First normalised data using fit_transform
- Fitted the data in model
- Then calculated cross_val_score

- `cross_val_score is:  0.7761950104741955`
- `Std is:  0.052237658601284105`
- Calculated **ROC AUC** score using inbuilt library functions
- `ROC AUC: 0.8438923395445134`
- inbuilt library functions
- Accuracy score:`0.8146067415730337`
- The shape of the ROC AUC curve is similar to what defined in theory



Part 4-

- Performed 5 fold cross validation using scratch made function
- `Mean accuracy: 0.7717196724433442`
- `Standard deviation: 0.0480414121926202`

Part 5-

- The only two continuous features in the data are “Age” and “Fare”.
- We know that it only makes sense to make contour plot only for the features which is continuous and not for those which has discrete finite values
- So we made contour plot using features as “Age” and “Fare” using plt.contourf
- The contour plot is about “Age”,”Fare” and the class
- In the graph the colour boundary represents that the 3d graph at that point are of equal elevation
- The contour plot is **elliptical shaped** which means that the features “Age” and “Fare” are dependent features, It is opposite to what **naive**

bayes assumption that features are independent, due to this only the accuracy of the Gaussian naive bayes is low

Part 6-

- The **accuracy of Decision Tree Classifier is higher than Gaussian Naive Bayes Classifier**
- `Scores is:  0.8110455151399734`
- `Std is:  0.01880179257025651`
- This is consistent with our contour plot result which signifies that the features are dependent whereas we selected Naive Bayes which assumes independent features.
- So the accuracy of the Decision Tree Classifier is high

Question 2:

Part a-

- Downloaded dataset using wget command called using os.system
- Load the dataset in df variable using pd.read_csv
- We hardcoded the columns names as it was not given in the dataset itself with the name **['Area','Perimeter','Compactness',Length of kernel','Width of kernel','Asymmetry coefficient','Length of kernel groove','Class']**
- Using matplotlib.pyplot.hist we plotted the histogram using bins=30

Part b-

- We find the prior probability of every class using sum(column==class)/len(column)

Part c-

- We convert df to X,Y
- We use bins=10 then discretized the X using np.digitize
- We take out the result using scratch only and not using any ML related library

Part d-

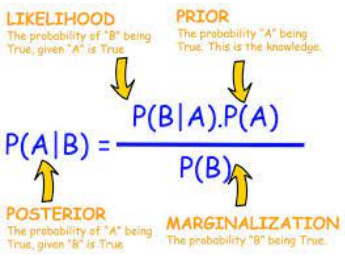
- We convert df to X,Y
- Then we calculate class conditional probabilities / likelihood using zeta=X[Y==class_value] then iterating over features for zeta then using these calculated likelihood/class conditional probabilities

Part e-

- We use **X_discretized** to plot the count of each unique element for each class
- We used bar plot here
- The plot comes out to be almost same as that of the distribution of samples, just the difference is that here we use X_discretized and also the plots here are per class so total plot here is 21 where it was 8

Part f-

- We find the posterior probability with the use of class condition probability calculated in part e using formula



- Then we used plt.subplot to plot posterior probability for each class and X_discretized and per feature
- At the extreme value of feature_value only one out of three classvalue is heavily dominating the other class values as shown in the figure
- As shown below in the figure this pattern is seen in all the features-classvalue plot