# Machine Learning Assignment -3 Student: Sowjanya Sunkavalli

ID: 700731896

# Programming elements:

Classification

### Question 1

1. Find the correlation between 'survived' (target column) and 'sex' column for the Titanic use case in class.

```
In [30]: | Import numpy as np # Importing numpy as np
import pandas as pd # Importing pandas as pd
import warnings # Importing warnings to exclude warnings
warnings.filterwarnings("ignore")

In []: | Itrain = pd.read_csv('train.csv') # Reading train data set as train dataframe
test = pd.read_csv('test.csv') # Reading test data set as test dataframe

In [17]: | Itrain['Sex'] = train['Sex'].replace(["female", "male"], [1, 0]) # Replacing Female and Male values with 1, 0
train['Embarked'] = train['Embarked'].replace(['S','C','Q'],[0,1,2]) # Replacing S, C, QA with 0, 1, 2

In [18]: | Itrain.fillna(train.mean(),axis=0,inplace=True) # Filling null values with mean

In [19]: | Itrain['Survived'].corr(train['Sex']) # Correlation of train for Survived and Sex
Out[19]: 0.5433513806577552
```

Imported numpy as np, pandas as pd and Warnings to ignore the warning statements.

Reading train data as train dataframe and test data as test data frame.

Replacing Female and Male values to equivalent 1, 0 in the Sex column using **replace function** And also replacing S, C, Q with 0,1, 2 values respectively in Embarked Column.

Filling Null values in train data set with mean values of corresponding columns using **mean**() function Finding the correlation value between Survived and Sex column using **corr**() function.

## a. Do you think we should keep this feature?

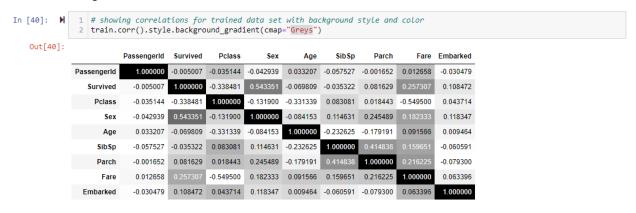
```
In [ ]: M 1 # Yes
```

Yes, I believe we should maintain this functionality. We may include this column when providing input to the model because we changed the values of the Sex column from female and male to 1 and 0

accordingly.

#### 2. Do at least two visualizations to describe or show correlations

#### Showing correlations



Finding the Correlations of the train data using corr() function and also here we can use style.background\_gradient(cmap="Greys") to provide the background style and color to the output visualizations.

#### 3. Implement Naïve Bayes method using scikit-learn library and report the accuracy.

```
In [21]: 📕
               1 X = train.drop(columns = ['Name','Survived','Ticket','Cabin']) # Dropping alphnumeric values
2 Y = train[['Survived']] # assigning Survived column
In [22]: H 1 # Splitting dataset into traning and testing data sets
                2 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 0)
In [23]: ► # Gaussian Naive Bayes
                  from sklearn.naive bayes import GaussianNB
                  classifier = GaussianNB() #creating the model
                   classifier.fit(X_train, Y_train) #Feeding training data to the model
               6 Y_pred = classifier.predict(X_test) # Predicting the test variables
               8  # Summary of the predictions made by the classifier
9  print(classification_report(Y_test, Y_pred)) # Classification report
               10 print(confusion_matrix(Y_test, Y_pred)) # confusion matrix
                   # Accuracy score
               12 from sklearn.metrics import accuracy score
               13 print('accuracy is',accuracy_score(Y_pred,Y_test)) # Accuracy of the model
                             precision recall f1-score support
                                                       0.79
                                                                   179
                  accuracy
                              0.79
0.78 0.79 0.78
0.80 0.79 0.79
                 macro avg
              weighted avg
                                                                    179
              [[90 20]
              accuracy is 0.7932960893854749
```

Dropping Name, Survived, Ticket, Cabin from training data set and assigning Data set to X using **drop** () function.

Assigning training Data Set containing only Survived column to Y

Splitting data set into training and testing data sets using train\_test\_split() method

Creating the model using Classifier = GaussianNB()

Feeding the training data to the model using **fit(data parameters)** function.

Predicting the test variables using **predict() function**.

Printing Classificationreport using **classification\_report()** function.

Printing Confusion Matrox using confusion\_matrix() function.

Getting Accuracy using accuracy\_score() function

#### Question 2

Implement Naïve Bayes method using scikit-learn library.

a. Use the glass dataset available in Link also provided in your assignment

Reading glass data file using read\_csv() method

Dropping Type columns and assigning remaining data set to X.

Assigning Type Column data set to Y.

2. Evaluate the model on testing part using score and classification\_report

```
In [31]: N
              1 # Gaussian Naive Bayes
                from sklearn.naive_bayes import GaussianNB # importing classifier
                classifier = GaussianNB() # assigning classifier with model
               4 classifier.fit(X_train, Y_train) # feeding training data to the model
               6 Y_pred = classifier.predict(X_test) # Predicting the dependant variable
               7 # Summary of the predictions made by the classifier
8 print(classification_report(Y_test, Y_pred)) # Classification Report
               9 print(confusion_matrix(Y_test, Y_pred)) # Confusion matrix
              11 from sklearn.metrics import accuracy_score
              12 print('accuracy is',accuracy_score(Y_pred,Y_test)) # Finding the Accuracy
                           precision recall f1-score support
                                         0.44
                                 0.19
                                                      9.27
                                 0.33
                                           0.16
                                                      0.21
                                        0.16 0.21
0.20 0.25
0.00 0.00
1.00 0.80
1.00 1.00
                                 0.33
                         6
                                 0.67
                                 1.00
                                                                   6
             accuracy 0.37
macro avg 0.42 0.47 0.42
weighted avg 0.40 0.37 0.36
                                                                   43
             [[4 3 1 0 1 0]
              [14 3 1 1 0 0]
               [3 1 1 0 0
                0 2 0 0 0
                0 0 0 0 2 0]
               [0 0 0 0 0 6]]
```

Creating the model using Classifier = GaussianNB()

Feeding the training data to the model using **fit(data parameters)** function.

Predicting the test variables using **predict() function**.

Printing Classificationreport using **classification\_report()** function.

Printing Confusion Matrox using **confusion\_matrix**() function.

Getting Accuracy using accuracy\_score() function

Implement SVM method using scikit-learn library.

a. Use the glass dataset available in Link also provided in your assignment

Reading glass data file using read\_csv() method

Dropping Type columns and assigning remaining data set to X.

Assigning Type Column data set to Y.

1. Implement linear SVM method using scikit library

```
In [37]: H

# Support Vector Machine's

from sklearn.svm import SVC # importing SVC

classifier = SVC() # creating the model

classifier.fit(X_train, Y_train) # feeding model with training dataset

Y_pred = classifier.predict(X_test) # predicting the dependent variable in the test dataset
```

2. Evaluate the model on testing part using score and Classification

```
# Summary of the predictions made by the classifier
print(classification_report(Y_test, Y_pred)) # printing the classification report
In [38]: ▶
                 \verb"print(confusion_matrix(Y_test, Y_pred))" \# \textit{printing the confusion matrix}
                  # Accuracy score
               5 from sklearn.metrics import accuracy_score # importing the accuracy_score
               6 print('accuracy is',accuracy_score(Y_pred,Y_test)) # printing the accuracy of predicted values with true values
                            precision recall f1-score support
                                          1.00
                                                     0.35
                                 0.00
                                           0.00
                                                     0.00
                                                                  19
                                 0.00
                                           0.00
                                                     0.00
                                 0.00
                                           0.00
                                                     0.00
                                 0.00
                                           0.00
                                                     0.00
                                 0.00
                                           0.00
                                                     0.00
                                                                   6
                 accuracy
                                                      0.21
                                                                  43
                                0.03
                                           0.17
                macro avg
                                                      0.06
                                                                  43
             weighted avg
                                0.04
                                           0.21
                                                     0.07
             [[ 9 0 0 0 0 0]
              [19 0 0 0 0 0]
                2 0 0 0 0
                               0]
                2 0 0 0 0 01
             accuracy is 0.20930232558139536
```

Importing svm from sklearn library.

Assiging model svc() to classifier

Feeding the training data to the model using **fit(data parameters)** function.

Predicting the test variables using **predict() function** 

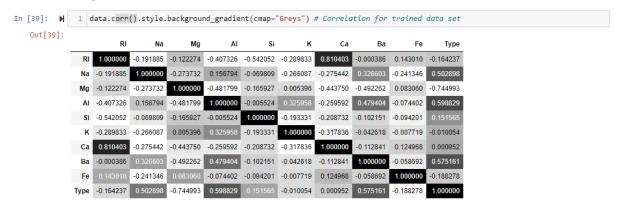
Printing Classificationreport using **classification\_report()** function.

Printing Confusion Matrox using confusion\_matrix() function.

Getting Accuracy using accuracy\_score() function

Do at least two visualizations to describe or show correlations in the Glass Dataset.

#### **Showing Correlations**



Finding the Correlations of the train data using **corr**() function and also here we can use style.background\_gradient(cmap="Greys") to provide the background style and color to the output visulalizations.

#### Which algorithm you got better accuracy? Can you justify why?

In []: N 4 After analyzing results got from training data with Naives Bayes and SVM model, from the above results of accuracy 2 # We can say Naives Bayes Algorithm is better than SVM 3 # Accuracy of Naive Bayes i.e 37.2% > Accuracy of SVM i.e 20.9%

**GitHub**: https://github.com/sunkavallisowjanya/MachineLearningAssignment3

Video Link: https://youtu.be/7bFOO9fPqQ4