CSCI 184 Homework 2
Due Date: May 1st, 2022

1) Naive Bayes – Cancer Tumor Classification

For this part, you will focus on a cancer dataset that comprises of 569 rows and 32 columns and perform Naive Bayes Classification.

What do you need to do?

1. Load the dataset from ‘cancer.csv’ into a pandas DataFrame and print it along with its shape. ‘diagnosis’ is the target variable.
2. Print the column names and the data type of each column.
3. Plot the ‘Radius Mean’ VS ‘Texture Mean’ along with the classes represented as colors or shapes. Is the data linearly separable?
4. Perform encoding on the target variable (here label encoding will suffice).
5. Divide the data into X and Y, where X is the set of features and Y is the target variable.
6. Split the data into train and test data. Choose a split size of 70 - 30.
7. Given the nature of the data and its features, choose which Naive Bayes is the most suitable. Mention this in your report along with why you chose the same. You may use the Naive Bayes from sklearn
8. Once you have trained your model, evaluate the model performance by printing the performance matrix.
9. Write a report with screenshots of your results and the final results for step 8.

10. Submit your code as an .ipynb file and a document reporting your findings.

2) Decision Tree - Diabetic Patient Classification

For this part, you will focus on building a Decision Tree using the ID3 algorithm for the diabetes dataset from Homework 1.

The link is here for reference:
https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv

What do you need to do?

1. Load the dataset. The link does not have any names for headers. Add them with the following (in order: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcome)
2. Print the DataFrame and its shape.
3. Divide the dataset into X and Y where X is the set of features and Y is the target variable. Here 'Outcome' will be your target variable
4. Split the data into train and test data. Choose split size as 70 - 30.
5. Train your Decision Tree model. You can use the Decision Tree from sklearn. By default, sklearn uses 'gini index' criteria for implementation.
6. Choose different values (2 to 10) for max_depth and calculate the train and test accuracy. Tabulate the results. [Use random_state = 1 for the Decision Tree]

Max_Depth	Train Accuracy	Test Accuracy
2		
3		
4		
:		
:		
:		
:		

Write a report with screenshots of your results and the final results for step 6.
Is there a pattern with the max_depth and train accuracy and max_depth and test accuracy? What do you think is the cause?

Submit your code as an .ipynb file and a document reporting your findings.
You may choose to use the same report for two of the parts, but please use two separate .ipynb files for the two parts.