Assignment 3:

So far you have practiced pre-processing on data and applying regressive methods. This assignment is focused on applying the following classification techniques:

1. Naïve Bayes
2. K-Nearest Neighbor (KNN)
3. Support Vector Machines (SVM)
4. Decision Trees (DT)
5. Logistic Regression (Logit)

NOTE: You may have already started applying Tutorial_6 _Classification.ipynb on your dataset you have used in your previous project. You may have started seeing some insights from the knowledge you gained from the domain. Share this in your report. For this Assignment, in addition to the work you have done on your dataset with the tutorial, please use the provided dataset given below and follow the instructions below:

Dataset and the description of the dataset to be used for this assignment can be found in files under labs→project3_Classification_Models folder.
The more datasets you use, the deeper insights you gain as each dataset is very unique.

Part 1.
A balanced and complete dataset is not necessarily good to use. There are still few important points to be taken care of before we create our ML classification learning model. For example, even if the dataset is preprocessed, if we don't know what features are useful, the model we create won't make much sense.
Your task is to find features that you think will be useful for your model in answering the question: What is the target class of the given observation?
This question judges your ability to differentiate and find the features useful for the questions you are answering.

Part 2:
A good train and test split is always useful. Duplicates in training and test set should be avoided. Any ML model that trains on a training set which helps explain a lot of variance in the data tend to yield higher accuracy on test set.
The task of this part is simple: Find a train-test split ratio (for example 80:20) that you think will be best for your ML model to be trained and tested on.

Part 3:
Practically, multiple different models are trained simultaneously (sometimes on different systems) and then compared. The model that explains the dataset the best and that yields the highest accuracy is generally chosen.
Your task is to calculate accuracy of all the models you create and tabulate it.

For your convenience, the following table is provided.

ML Model	Accuracy on Test Set (Provide accuracy in %)
Naïve Bayes	
KNN	Also provide the K value for which you got the highest accuracy
SVM	
DT	
Logit	

NOTE: For KNN when you create a model, try using different values for K (neighbors) and choose the K that gives the highest accuracy. Provide the accuracy of the model that gave the highest accuracy and also the K value.

Part 4: Provide a one- or two-page report on important decisions you took or observations you made. Include any visuals in case you have used them for your analysis.

Accuracy measures:
The most common accuracy measure is given as:

$$Accuracy\ Rate = \frac{Number\ of\ Correctly\ classified\ observations}{Total\ number\ of\ observations}$$

$$Accuracy\ in\ \% = Accuracy\ Rate * 100$$

Either implement the accuracy formula on your own (a quick and neat idea would be to use confusion matrix) or use sklearn library.
A well-defined procedure already exists in sklearn library. Use the following syntax:

```
from sklearn.metrics import accuracy_score
```

Even confusion matrix comes prepacked with sklearn.

```
from sklearn.metrics import confusion_matrix
```

Read more about accuracy_score:
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

Confusion Matrix is practically used to find various different measures including accuracy.
https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

Know more about confusion matrix library in sklearn
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html