NAME: OMAR USMAN UPPAL

CMS ID: 203515

AI/ML ASSIGNMENT 1

TASK 1:

Task 1 is about reading the "Tehsil Schools.xlsx" file and answer three questions:

Dataset has 86 data rows and 14 columns. Each column has heading which was printed in the code and python code was written accordingly. Pandas library is used to find answers.

Python File for Task 1: *"Task1_Tehsil.py"*

Answer 1: The names of all schools, for which ZERO number of students passed in 10th class exam, are:

'1. GGHS Bara Ghar', '2. GGHS Karyal Chak No.17', '3. GGHS 14/66', '4. GGHS Anand Garh Chak 8', '5. GGHS Chander Nager', '6. GHS Shahkot College Road', '7. GHS Sangla Hill', '8. GGHS Hanjali', '9. GGHS Badhu Malhi', '10. GGHS Islam Nager', '11. GGHS kartar Pur'.

Answer 2: There are **42** large sized schools which is **48.84%** of total schools.

Answer 3: "GHSS More Khunda" school has the highest % drop out (74.59%) amongst the large sized schools.

TASK 2:

Python File for Task 2: *"Task2_NBC_Formula.py" (for NBC formula) and "Task2_NB_Scikit.py" (for Sklearn workings)*

My CMS ID is 2035**15** therefore the dataset I used is "6_car.csv" as 15%21=**6**.

Initial observation: There are 1728 samples with 21 attributes. There are 4 classes (0, 1, 2, 3). The attribute values are 1 or -1.

Naive Bayes Classifier using NBC formula:

First I implemented the NBC from scratch to develop the understanding. Top 80% of the samples are selected as **training set (1382)** while last 20% are **test set (346)**.
P(Class = j | attribute =1 to 21) ∝ P(Class=j) * P(attribute 1|class=j)*…*P(attribute 21|class=j)
Where j is {0,1,2,3}. We calculate different values for right hand side of **"∝"** using **training set** and then used those values to predict the *CLASS* of **test set**.
Following results were achieved using the above method:

- **Accuracy of predicting the class of training set was 92.91%**
- **Accuracy of predicting the class of test set was 69.94%**

Bernoulli Naive Bayes Classifier using Sklearn:
As first step we used the same data samples as in our first experiment to compare the two results. We have used "BernoulliNB" type to check the result as our dataset has 2 discrete values only. The results using Sklearn library are:

- **Accuracy of predicting the class of training set was 93%**
- **Accuracy of predicting the class of test set was 70%**

There was only 1 more value that was predicted correctly using the sklearn library than the classifier built by our self.

Bernoulli Naive Bayes Classifier using Sklearn and Random samples:
In the first two experiments we had split the data set which was not random. Initial 80% of sample was selected which may not be a good criterion. Therefore in this experiment we will use "train_test_split()" module of Sklearn library to split our dataset and see the results. Our sample split ratio still remains the same as 80% train and 20% test. The results are:

- **Accuracy of predicting the class of training set was 87%**
- **Accuracy of predicting the class of test set was 86%**

The results show that now the accuracy of "test" set has improved but "training" set has reduced. On further investigation and more trials by changing the value of **"random_state"** we found that the accuracy of "training data" does not exceed 89% while accuracy of "test data" fluctuates between 85%-89%.

Observations:
If we look at the data samples than we can see that out of 1728 samples Class 0 samples are 70% of the data while when we select top 80% samples than Class 0 constitute 73% of the training data while ratio of other classes has less representation. This means that 80% samples selected do not have the true representation of all classes as even after split the 20% test set has more representation of Class 2 and Class 3.

	Total	80%	20%
Class 0	1210	1012	198
Class 1	384	321	63
Class 2	69	23	46
Class 3	65	26	39

In the case of random sampling the distribution would be different and the training data sample would have better representation of other classes and so the accuracy in case of test data increased.