**Naïve Bayes for Demo weather data**

In the Naive Bayes classification we first tried to load the data using the load\_data method. As we have three .csv files we used this function to load these three files we used special module csv to do that. This functions outputs the features and the label separately list of lists and a list respectively. Check the comment in the main code file (dataloader.py).

We have also created a validator function that validates our prediction and also a test function to test the file. Those are our evaluators

In our texturized method we mapped the existence of one feature in the given input feature list into binary number 0 and 1. We do it for all the columns of continent, season, location and weather. The wind speed is by itself has a binary values high and low so we can set 0 and 1 accordingly. Through training function we have come up with the probability for our label (prior probability) and the probability of the features

We have put a comment as much as possible in the implementation to make it clear for the readers. And In order to run the code we have provide main.py file where every functions are imported and loaded

**Logistic regression for Demo weather data**

Here also we have the same just like In the Naive Bayes classification we first tried to load the data using the load\_data method. As we have three .csv files we used this function to load these three files we used special module csv to do that. This functions output a list where the data are separated using comma. Check the comment in the main code file (dataloader.py).

We have a method called clear\_data(). This function converts specific columns in the data from their original values to their corresponding indices based on unique values in those columns. Additionally, it converts the temperature values to their corresponding indices. The modified data is then returned.

And then through a function called evaluate which takes the test\_data, weight and bias we evaluates the accuracy of a binary classification model by comparing the predicted outputs with the true labels for each row in the test data. It counts the number of correct predictions and calculates the accuracy as a percentage.

In the main category we used functions for calculating the sigmoid activation, training a logistic regression model using gradient descent, and making predictions. Using the learned model, we believed these functions can be used together to perform binary classification tasks based on the logistic regression algorithm.

In order to run the code we have provide main.py file where every functions are imported and loaded