CT5165 Assignment 1

There are a total of 8 open-source machine learning packages available which are Tensor Flow, Keras, Scikit-learn, Microsoft Cognitive Toolkit, Theano, Caffe, Torch, and Accordnet. The purpose of every package is different. TensorFlow and PyTorch library are majorly used for deep learning and Neural Networks. Keras library supports both convolutional and recurrent networks whereas Scikit learn focuses on data mining and analysis.

- Justification:

1. For the given classification task, I have using a Scikit-learn package over another machine learning packages.
2. The entire package is heavily written in python, and it becomes very easy to understand and implements the various algorithms. The scikit-learn library helps to split the entire dataset into train and test datasets and provides different machine learning algorithms to solve different problems like Classification, Clustering, Regression. Also, it provides different algorithms such as model selection, Dimensionality Reduction, and Pre-processing. It provides a wide selection of supervised and unsupervised algorithms. This package is composed of three main libraries which are NumPy(mainly used for Scientific Computation), Pandas (to process Tabular Data), Geo, and Scipy(for sparse matrices). Within few lines of code, it is easy to accomplish classification and many other tasks.
3. I have used sci-kit learn's Count Vectorizer library in the SpamDetectionFilter NLP project. Being a beginner, it's package documentation which includes narrative documentation, installation instructions, class references, implementation of functions helped me to understand the algorithm well and hence I found this package very helpful and convenient.
4. Due to previous experience, easy to use, simplicity, and robustness I think Scikit learn package is suitable for the given classification tasks.

- Data Import and Data Pre-processing:

1. It is the process of removing incomplete, corrupt, incorrect, and irrelevant data to produce the exact correct accuracy of any task.
2. Before moving ahead in the given classification, first in order to standardize the data, I followed the below steps for data cleaning and preparation.
   a. Converted supplied dataset file from .txt to .csv.
   b. Read the CSV file using the read_csv()method and passing file path as a parameter of the function, dataset_train variable used to store the training dataset.
   c. To check there are any NULL values in the training dataset by using is.na() method. Luckily, there is no null values present in the dataset.

d. There is a total of 10 columns in the dataset and 154 rows. Except for the fire column all other columns contents are in numeric format. To make the dataset standard, copying the entire fire column in the y_train variable and checking its unique values.

The fire column consists of 7 unique elements (.unique()) such as

['no ', 'yes ', 'yes', 'yes ', 'no', 'no ', 'no '].

e. To remove the white space from the start and end of the string, I have used str.strip() method and again checking for unique values and storing them into the dum_train variable.

['no', 'yes'].

f. The y_train column has string values, the machine learning model doesn't process the string values hence using pandas get.dummies function to convert them and storing them into dum_train variable. So, all the data values of dum_train are numeric.

g. Drop the existing fire column from dataset_train and storing the dataframe (which has 9 columns) into x variable.

h. Picking a 'No' column from dum_train variable and saving it into y1_train.

i. After following all these steps, to develop a model all required data are pre-processed, standardized, and stored into different variables.

- Classification using Naïve Bayes –GaussianNB algorithm:

1. GaussianNB algorithm is a type of Normal Distribution.
2. It is majorly used when features have continuous data, by assuming that continuous data associated with each class is following gaussian distribution.
3. Gaussian Distribution :

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

4. This model simply finding the mean and standard deviation of the points within each class. At every data point in the dataset, the datapoint and mean of that class is subtracted and get divided by standard deviation of that class.[1]
5. In the given dataset, the goal of the classifier is to predict **fire** based on the other attribute. The training and testing dataset is already split into wildfires_training.txt and wildfires_test.txt respectively. (Used training dataset to develop classification model.)
6. Post completing the data preprocessing, implementing the Machine learning Gaussian Algorithm on it. Importing Naïve Bayes GaussianNB library and saving the model into a model variable.
7. To develop the classification model with fit() method. In the supervised learning fit() method takes 2 arguments x = dataset_train and target = y_train i.e <modelname>.fit(x,y)
8. To make predictions using predict() function with the train set as its parameter and storing it into the y_pred_train variable.

9. To test the quality of a developed model, Comparing the model's prediction with the target values for the training dataset. Importing accuracy score library from sklearn.metrics.
   By the following method calculate the accuracy of the model :
   accuracy_score(y1_train, y_pred_train).
10. The model is giving an accuracy of 90.90% with class 'no' on train dataset. Followed the above data preprocessing steps on the test dataset.
11. Test data consists of 49 rows and 10 columns. All the column data is in integer format except the fire column, but the fire column of the test data has only 4 unique columns such as
   ['no ', 'yes ', 'yes ', 'no '].
   Performed the same steps for developing a model and predicting the result.
12. The model is giving an accuracy of test data is 88.00% with class 'no'

- Classification using KNN–K Nearest Neighbor algorithm:

1. KNN algorithm is a supervised learning algorithm that is suitable for both classification and regression.
2. Uses the similarity feature to predict the group that the new point will fall into.
3. It finds the distance between the query points and all the dataset points. It looks at K closet query points in the dataset points whatever is the majority class in the group of K data it is the predicted class of algorithm. K in the KNN defines the no of neighbour.eg if K=3 then it finds the 3 nearest neighbor and calculates the difference [2].
4. Distance can be calculated using 'Euclidean distance', 'Manhattan distance', 'Hamming Distance', and 'Minkowski Distance'.
   Euclidean distance can be calculated as –
   Distance between (x1,y1) and (x2,y2) = $\sqrt{(x1-x2)^2 + (x2- y2)^2}$
5. In the classification task, I have used the Euclidean distance method and predicting the fire based on the other attributes.
6. Pre-processing the training data set and importing KNeighborsClassifier from sklearn.neighbors. Developing a KNN model with a fit() method and have passed trained dataset at x-axis and fire column at y-axis with class – no
7. Predicting the result of the model, bypassing preprocessed data(followed above explained preprocessing steps)as an argument.
8. The model is giving an accuracy of train data is 86.36% with class 'no' on dataset_train.
9. Followed the above data preprocessing steps on the test dataset and developing a test model. To make predictions used predict() method and to test the quality of the model using the accuracy_score() method.
10. The developed model is giving an accuracy of 82.00% with class 'no' on dataset_test.

- Observation of Multinomial and KNN:

1. The accuracy of the train data set of both the models are not having a huge difference. They are quite similar with a difference of 4.54 %.
2. KNN and GaussianNB algorithms both are easy to interpret the prediction.
3. GaussianNB algorithm simply calculates the mean and standard deviation of the class and based on the formula calculates the probability of data point of each class in the developed model.
4. Whereas in KNN, it finds the nearest neighbor of query data point and assigns the class/labels based on the nearest neighbor's value.
5. The accuracy of both the models is 96.46% similar, as data preprocessing steps performed on both the models and train dataset are the same. In KNN the accuracy varies as the k values changes.
6. KNN is sensitive, due to noise and irrelevant data the accuracy gets impacted.

- References :

[1]https://iq.opengenus.org/gaussian-naive-bayes/
[2]https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification