CT4101-Machine Learning Assignement-1

Name : Saikrishna Javvadi
Student-ID : 20236648
Class : MSc in Computer Science – Data Analytics

Package of choice and it's features:

I would be using the Scikit-learn ML Framework for my assignment because Scikit-learn provides a good number of useful utilities for splitting data, data manipulations, metrics functions, artificial dataset generations, and so on. Scikit-learn has good documentation, and a clean, consistent API and it is also easier to get help on the internet if you are stuck somewhere while programming because of the huge community support online, which is due to it's open source nature. Another main advantage of this package is it is easy to access and a good choice for simpler data analysis tasks . [1]
Some of the main features of this package are:
1) It is built on NumPy, SciPy, and matplotlib, so it integrates well with all these libraries.
2) Open source, commercially usable.
3) Scikit-learn is reusable in various contexts and it is accessible to everybody.
4) A number of pre-processing steps like taking care of missing data , encoding categorical data and feature scaling could be easily achieved by the use of scikit-learn.
5) Using this package, numerous model selection steps like splitting the dataset into training and test sets , tuning the hyperparameters of an estimator etc could be done easily.
6) Scikit-learn implements a lot of non-neural net based algorithms that are commonly used in data science like classification , regression and clustering.
7) Using Dimensionality reduction in scikit-learn, the number of attributes in data can be reduced for further visualization, summarization, and feature selection.

Data Pre-processing:

Since the data provided for the assignment was already divided into training and test sets , and doesn't have any missing data, the data pre-processing step was a bit easier. Firstly, the data provided in tab delimited txt file was converted into a excel file with .xlsx, so that it can be visualized properly and for the ease of use. Later, these excel files(both training and test) were loaded into a pandas dataframes in Jupyter Notebook . Since the dependent variable "style" was a categorical variable, using the LabelEncoder class of sklearn.preprocessing this variable was encoded to numerical values. The "beer_id" variable had no significance/impact on the dependent variable , so this variable wasn't used while performing the feature scaling in the next step and later on while training the model. The above training and test dataframes are converted into numpy arrays to fit into the models. Before fitting the data into the models for training, using the StandardScaler class of sklearn.preprocessing , a feature scaling step was done to reduce the impact of a single variable on Bayesian distance. This step helped me significantly in increasing the accuracy of model further.

Algorithms used & their Description:

The two algorithms that I used for this classification task are K-Nearest Neighbours and logistic regression. Both the algorithms are described below as per my understanding:

K-Nearest Neighbours:

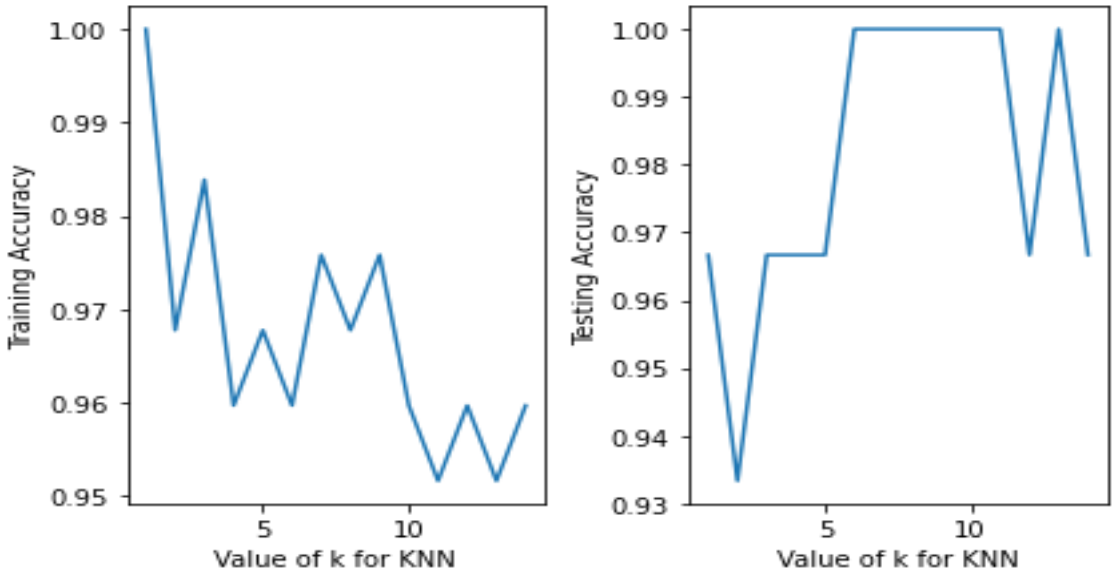
This algorithm computes the distance between the new data point with every training example present. For computing the distance, measures such as Euclidean distance, Hamming distance or Manhattan distance can be used. The model picks K entries in the dataset which are closest to the new data point, then it does the majority vote i.e the most common class/label among those K entries will be the class of the new data point. Our task is to build a KNN model which classifies the new beers into ale, lager or stout based on the given features/attributes.[2]

Logistic regression:

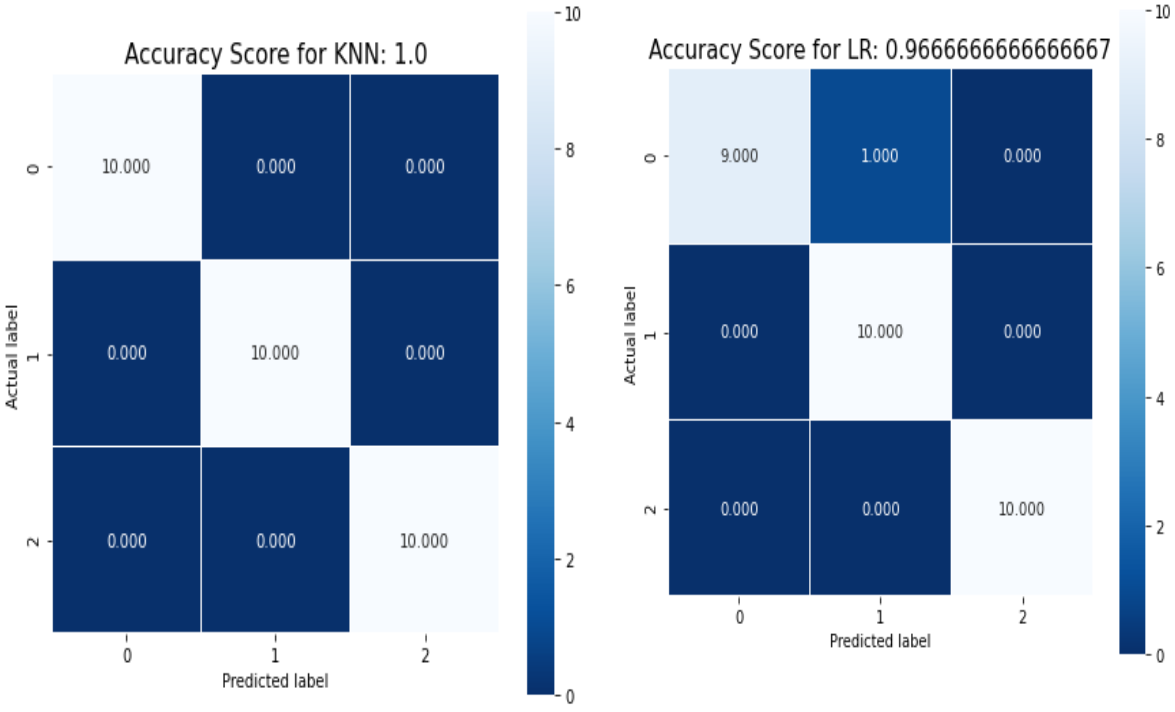
Logistic regression is a classification algorithm rather than a regression algorithm. It can be used when the dependent variable(target) is categorical unlike linear regression which is used when data is a continuous value. Basically, it measures the relationship between the categorical dependent variable and one or more independent variables by estimating the probability of occurrence using its logistics function i.e For any given training example x, we should calculate $\hat{y}$ which is equal to $P(y=1|x)$.Logistic regression is named for the function used at the core of the method, the logistic/sigmoid function. Logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes. One could also say, logistic regression is most basic form of a neural network with only one neuron that is can be used for binary/multi-class classification.[3][4]

Report on the results of the models:

Firstly, Here is the link to the code implemented for training and testing the models https://github.com/saikrishnaj97/beer_style_predictions. The Algorithms were trained using the beer_training and tested on beer_test as per the instructions in the assignment. Firstly talking about the KNN algorithm, a training set accuracy of 97% and a test set accuracy of 100% was observed for KNN with k=7, which was the model that I finalized on after testing on a range of k-values. Also, the test set accuracy of 100% was observed for all the k-values ≥ 6 and ≤ 10 . Firstly, The data was fit into the KNN algorithm using the KNeighborsClassifier class of sklearn.neighbors for a range of k-values from 1-15 , the different accuracies observed over these range of values are plotted below for the train and test set using matplotlib.pyplot. From the figure, It can be seen that an accuracy of 100% was achieved for the k-values between 6 and 10 on the test set, It can also be observed that even the k-values from 1-5 also have pretty decent accuracy i.e 93%+. Although, for k-values≥ 10, we can observe the accuracy being varied which might be the case due to less amount of data, so I choose my k-value as 7 and fit the data into that finally because it lies between k-values(6,10) for which I observed 100% accuracy. I should also mention that, I observed this significant increase in the accuracy of KNN model(from ~55% to 90%+)after performing the feature scaling step which was mentioned previously.



Most of the steps involved in implementing the logistic regression algorithm are same as the KNN implementation mentioned above. Firstly the pre-processed data was fit into the logistic regression model using the LogisticRegression class of sklearn.linear_model library. There were few parameters to be adjusted while fitting the model for which I referred to the logistic regression documentation of scikit-learn[4]. Doing which an accuracy score of 99% and 97% was observed for this model on the training dataset and test set respectively. The confusion matrix of accuracy on the test sets for both the algorithms used is plotted using heatmap of seaborn library as shown below.



Interpreting the Results:

The two algorithms used (KNN and Logistic Regression) for the assignment give very similar results in terms of accuracy of the models. Since the data that we are using to analyze the results on(test data) is considerably less (around 30 records) , a very high accuracy of the models is observed, i.e while KNN for k-values ≥6 and ≤10 gives a 100% accuracy ,logistic regression model gives and accuracy score of around 97% as shown in the figures above. So we can say that KNN algorithm does a very good job predicting the style of the beer with all the given attributes. If one could collect more data to test and train the algorithms on , we can clearly distinguish the performance of each model . Scikit-learn makes the process easier by saving time for Data Scientists by providing functions to directly pre-process the data and fit it into the models, without having the hassle to implement the algorithms from scratch.

References:

1) Scikit-learn : https://scikit-learn.org/stable/
2) KNN: https://towardsdatascience.com/knn-using-scikit-learn-c6bed765be75
3) Logistic Regression:
https://www.tutorialspoint.com/scikit_learn/scikit_learn_logistic_regression.htm
4)Logistic Regression :
https://scikitlearn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html