Epoch Hackathon Report

Nelakuditi Rahul Naga - AI20BTECH11029

1 Data Preprocessing

I have found that the given red wine dataset has no null values using pandas and all the attributes are of float type. The label quality is of integer type. I did not find much use for extensive data preprocessing since the given dataset is very clean.

2 Exploratory Data Analysis

I have first uploaded the given data into a training dataframe using pandas. I have then explored the given red wine dataset by using various builtin functionalities in pandas. I have also plotted various features/attributes against quality of the wine. Clearly, quality variable is a **binary** label in the given dataset. So, we are essentially dealing with a binary classification problem in supervised machine learning. I have included all the relevant plots of data analysis in the .ipynb file that I have submitted.

3 Feature Engineering

I have interchanged the quality and n_value columns in the training dataframe so that quality feature (which is the label) is the last column in the dataframe. I have designed my machine learning models in accordance with this change. I have found from the exploratory data analysis (and from some hit and trail) that dropping the attributes k_value, l_value, m_value, n_value and percentage_free_sulphur from the training dataframe significantly improves the accuracy of my models which I have briefly discussed below.

4 Algorithms used to solve the problem

I have implemented two different machine learning algorithms from scratch to solve the given problem. I have explained the code snippets in my algorithms by writing comments

wherever necessary in the .ipynb file that I have submitted. The brief explanation of the algorithms and the **final accuracies** that I have achieved using them are as follows :

4.1 Decision Tree

I have implemented a decision tree using binary univariate split and entropy as a impurity criterion. I have also written appropriate functions to evaluate the model using k-fold cross validation. I have chosen k = 5 since in every fold the train-set : test-set ratio is required to be 80 : 20 as mentioned in the problem statement. Then I have reported the final accuracy as the mean accuracy obtained after all the 5 folds have been utilized as a test set. The final accuracy that I have obtained is **75.862%**.

4.2 Random Forest

I have implemented the Random Forest algorithm to try and improve the classification accuracy on the given dataset. I have implemented this algorithm by bagging 100 decision trees that use binary univariate split and gini index as a impurity criterion for classification. I have utilized some functions that I have written to implement the decision tree mentioned above in my Random Forest algorithm. I have used the train_test_split functionality from the sklearn library to divide the given dataset into train and test sets with the ratio 80 : 20. The final accuracy that I have obtained on the test dataset is **84.688%**.

5 Conclusion

I have tried out two different algorithms to solve the given classification problem. I was able to achieve a good jump in accuracy of classification on the test dataset using the Random Forest algorithm.