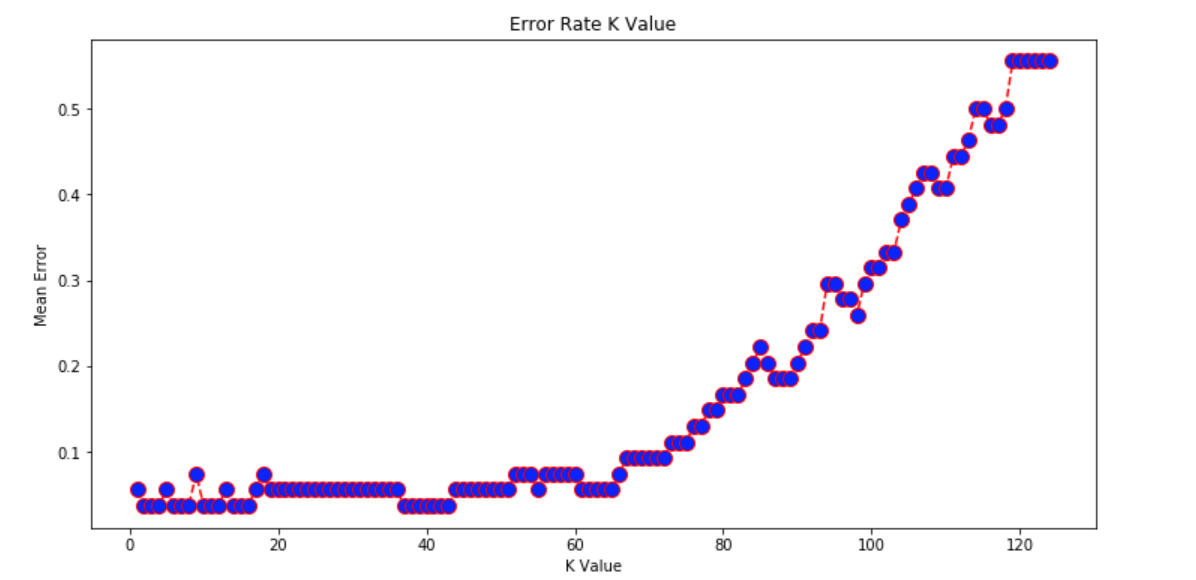
**INTRODUCTION**

Machine learning plays a significant role in the present times. It is a branch of artificial intelligence and is used for analyzing the data. Machine learning took birth many years ago but has gained momentum with the increasing Big Data. The main aim of machine learning is to make a machine learn from the data so there can be less human intervention. This is possible through various algorithms which are nothing, but a set of rules used in solving the problem by a computer. There are various types of machine learning algorithms used to analyze and predict the data. Some of them are Linear Regression, Decision Tree, K-Means, Random Forest, KNN (K nearest neighbors) etc.

In this assignment, K nearest neighbors (KNN) algorithm is used on a wine dataset. K nearest neighbors algorithm is a simple algorithm used for both classification and regression. It is a non-parametric algorithm and is widely used because it is easy to interpret and understand it. It is a lazy algorithm which classifies new cases on the basis of previously stored cases. It has many industrial applications and usages. Some of them can be medical data mining, recommendation systems, facial recognition, anomaly detection, loan management, credit rating, forest inventory evaluation, fingerprint detection etc.

In this part of the assignment, KNN is performed through Python in Jupyter notebook. Firstly, the necessary packages are imported, and the data file is extracted. KNN algorithm majorly includes the five steps i.e. Preprocessing (Loading and Target), Splitting into Train and Test Sets, Feature Scaling, Fitting the model and making Predictions. After performing these five steps, results are evaluated.

In this dataset, the prediction in question is regarding the type of Wine whether Class 1, Class 2 or Class 3 type. Here the target values are types of Wines. Thus, y variable contains Wine types column and X variable contains all the columns except the wine types column. After the Split into Train and Test followed by Feature Scaling, the model is fitted, and predictions are made. While fitting the model, the value of K is taken to be 5.



A loop from 1 to 125 is executed to calculate the mean error and a graph reflecting Error Rate K Value is plotted. It can be seen that when K=3, the mean error is lowest reflecting the highest accuracy of the model. The following table reflects different accuracy rate at different K values. The values may change each time the code is run.

|  |  |
| --- | --- |
| K value | Accuracy Rate |
| K=3 | 96.30% |
| K=5 | 94.44% |
| K=60 | 92.59% |
| K=90 | 79.63% |
| K=102 | 66.67% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | K=3 | K=5 | K=60 | K=90 | K=102 |
| Average Precision | 0.97 | 0.95 | 0.94 | 0.83 | 0.50 |
| Average Recall | 0.96 | 0.94 | 0.93 | 0.80 | 0.67 |
| Average F1-score | 0.96 | 0.94 | 0.93 | 0.77 | 0.57 |
| Average Support | 54 | 54 | 54 | 54 | 54 |

The above table reflects the average values of Precision, Recall, F1-score, and Support. It can be seen that the average value of Precision and Recall is higher when K=3 reflecting it to be the optimum value. Also, the confusion matrix reflects high performance when K=3.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted Type 1 | Predicted Type 2 | Predicted Type 3 |
| Actual Type 1 | 16 | 0 | 0 |
| Actual Type 2 | 1 | 22 | 1 |
| Actual Type 3 | 0 | 0 | 14 |

The above table implies that all Type 1 and Type 3 wines were predicted accurately. It further reflects that out of 24 actual Type 2 Wines, 22 were predicted accurately while the remaining 2 were predicted as Type 1 and Type 3. This table also highlights that the accuracy of the model is maximum when K=3.

**CONCLUSION**

This brings us to the end of the report. On the basis of the analysis, it can be said that the KNN algorithm is a lazy algorithm which is non-parametric. It is commonly used because of its easy interpretation and understanding. It is very helpful in non-linear data and can be used for classification as well as regression. But it has some disadvantages in form of speed and money. It is very slow and requires huge space or memory for storing the training data resulting in an increase in the cost. Also, it relies on the Euclidean distance which is very sensitive to different magnitudes and may give bias results if the data is not properly scaled.

In the present assignment, data normalization is done to get unbiased results. Later the model is stored in PKL file format so that model can be deployed as an API endpoint. The data and prediction results are attached along with app.py file which contains the code for deploying the model as an API endpoint. Thus, it can be concluded that the K nearest neighbors algorithm is instance-based learning algorithm. Moreover, with Scikit-learn, it has become easier to perform it. But despite its several advantages, it still can not be used with multi-dimensional or large-dimensional data.

**REFERENCES**

* Sunil Ray. (2017, September 9). Essentials of Machine Learning Algorithms (with Python and R Codes). Retrieved from <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>
* Avinash Navlani. (2018, August 2). KNN Classification using Scikit-learn. Retrieved from <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>
* SAS. (2019, February 24). Machine Learning: What it is and why it matters. Retrieved from <https://www.sas.com/en_us/insights/analytics/machine-learning.html>
* Scott Robinson. (2018, February 15). K-Nearest Neighbors Algorithm in Python and Scikit-Learn. Retrieved from <https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/>