**CLASSIFICATION USING NEAREST NEIGHBORS**

NORTHEASTERN UNIVERSITY - SILICON VALLEY

SUBMITTED TO: SUBMITTED BY:

YIQING WANG ANUBHAV RASTOGI

(PROFESSOR) (STUDENT)

**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.

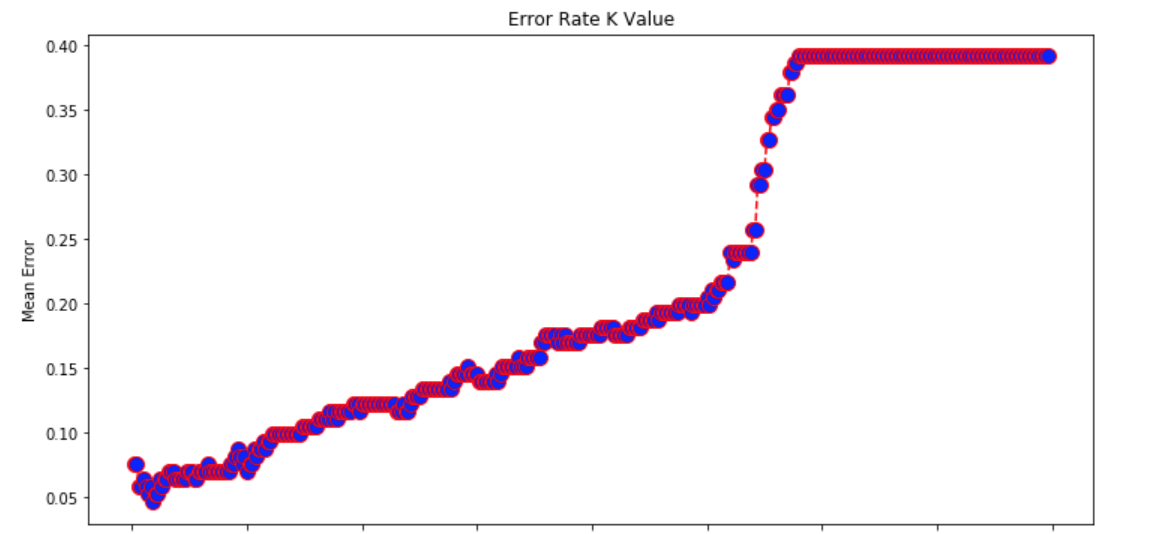
The present assignment talks about K nearest neighbors (KNN) algorithm. K nearest neighbors algorithm is a simple algorithm used for both classification and regression. It is a non-parametric algorithm and is widely used because it is easy to interpret and understand it. It is a lazy algorithm which 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.

The present assignment has two parts: Part A and Part B. Both the parts involve different datasets for carrying out the K nearest neighbors algorithm. Part A involves Breast Cancer Wisconsin (Diagnostic) dataset and Part B involves Wine dataset. The same set of rules/algorithm is run on both to find out the results.

**PART A**

In Part A 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 present dataset, we aim to predict the diagnosis of Brest Cancer whether M (Malignant) or B (Benign). So here, the Diagnosis column is the targeted values. The y variable contains the diagnosis column and X variable contains all the columns except id and diagnosis columns. The id column is irrelevant here, so it is not included while setting X and y variables.

After preprocessing, the next step i.e. Train Test Split is performed so as to avoid the problem of over-fitting. Here we have split the data into training and test set in the ratio of 7:3 i.e. 70% of the records are contained in training dataset and the remaining 30% of the records are contained in the test dataset. The third step involves feature scaling which is very important to make a uniform evaluation. Normalization is generally considered a good practice in machine learning because the data values may have a widely varying range. So, it is necessary to scale the data so as to avoid biased predictions. It is done with the help of **StandardScaler()** function imported from **sklearn.preprocessing** library. Now, the next step is to fit the model which is done by importing **KNeighborsClassifier** from the **sklearn.neighbors** library and fitting the parameter **n\_neighbors** = 5. Although the ideal K value is known only after testing and evaluation, 5 is generally used to initiate the algorithm. The last step is to make the prediction which is done through the **confusion\_matrix** and **classification\_report**. Also, **metrics.accuracy\_score** is printed to reflect the accuracy of the model.



In order to find out the optimum K value, a graph reflecting the mean error of the predicted values for all K values is plotted. The K value with lower mean error is considered to be the most optimum K value. This is done by executing a loop from 1 to 399 wherein mean error is calculated in each iteration appending result to error list. These error values are plotted against K values depicting the Error rate K value. It can be seen from the plotted graph the optimum K value is equals to 3 as the accuracy rate is maximum at this value. The following table reflects the different K values along with the accuracy rate. The following values may change each time the code is run.

|  |  |
| --- | --- |
| K value | Accuracy Rate |
| K=3 | 94.15% |
| K=5 | 93.57% |
| K=60 | 90.64% |
| K=90 | 88.30% |
| K=152 | 85.96% |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | K=3 | K=5 | K=60 | K=90 | K=152 |
| Average Precision | 0.94 | 0.94 | 0.92 | 0.89 | 0.88 |
| Average Recall | 0.94 | 0.94 | 0.91 | 0.88 | 0.86 |
| Average F1-score | 0.94 | 0.93 | 0.90 | 0.88 | 0.85 |
| Average Support | 171 | 171 | 171 | 171 | 171 |

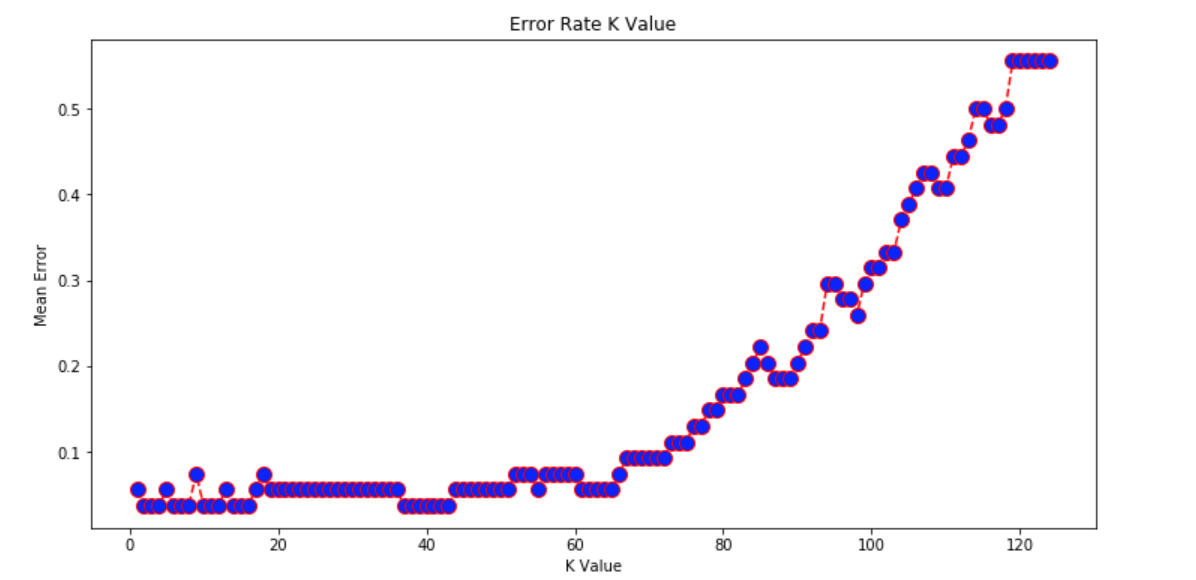
Precision is the depiction of true positive instances reflecting the relevancy. It is the fraction of retrieved instances which are relevant to the query and is maximum when K is 3 and 5. Recall reflects the relevant instances which were retrieved successfully and in the present case, it is maximum when K is 3 and 5. F1 score is the harmonic mean of precision and Support represents the number of samples of the true response present in the respective class. The F1-score when K=3 is slightly higher reflecting that the best optimum value of K is 3. Also, the confusion matrix which describes the performance of the model is best when K=3.

|  |  |  |
| --- | --- | --- |
|  | Predicted Benign | Predicted Malignant |
| Actual Benign | 102 | 2 |
| Actual Malignant | 8 | 59 |

The above table implies that out of 104 actual benign diagnoses, 102 were predicted as benign and 2 were predicted as Malignant. It further reflects that out of 67 actual malignant diagnoses, 59 were predicted as malignant and remaining 8 were predicted as benign. This reflects that the accuracy of the model is maximum when K=3.

**PART B**

Part B of the assignment includes the same machine learning algorithm i.e. KNN on a different dataset which is about Wines. In this dataset, the prediction in question is regarding the type of Wine whether Class 1, Class 2 or Class 3 type. This part of the assignment is also performed through Python in Jupyter notebook. The steps and code for performing KNN on this new dataset are similar to the previous one. After importing the relevant libraries along with the data extraction, the same five steps of KNN algorithm are performed. 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, normalization is done in both the parts so as to get unbiased results. However, both the datasets reflect the same optimum K value. Part A of the assignment which contains the Breast Cancer Wisconsin (Diagnostic) dataset predicts the Diagnosis whether Benign or Malignant. On the other hand, in Part B of the assignment, the KNN algorithm is performed on Wine dataset to predict the different type of wines. On comparing the performance of KNN algorithm on both datasets, it has been found that KNN algorithm functions well on wine dataset reflecting 100% accuracy most of the time when it is run. But here it is reflecting the instance where the accuracy is highest to 96.30%.

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/>