**Module 1 Predictive Analytics**

**Author’s Name:** Shivani Sharma

**Course Number:** ALY6020

**Course Title:** Predictive Analytics

**Academic Term:** Fall 2019 CPS Analytics

**Instructor’s Name:** Marco Montes de Oca

**Assignment Completion Date:** 09-22-2020



**Introduction**

This week 1 assignment of predictive analytics is based on the problem statement of image classification using “k nearest neighbor’s classifier” algorithm on Fashion MNIST dataset. This dataset consists of different clothing items which are identified with 0-9 numeric labels.



Before starting with the analysis, I would like to explain what is “K nearest neighbor” classification algorithm. It is a supervised machine learning algorithm designed for dividing the data into categories based on the features.

There are mainly following 3 steps for implementing this algorithm:

1. Calculate Euclidean distance: In simple terms we find out the distance between 2 rows of the data set. Here is the formula:

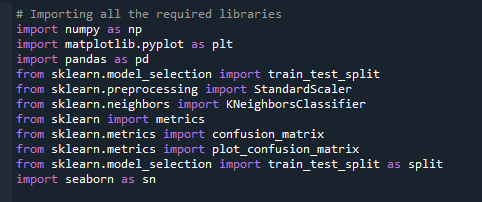


1. Get Nearest Neighbors: Then we calculate distance for each data point and find out the subsets for the closest neighbors for all the data points.
2. Finally, we do the prediction for the test data set based on the k nearest neighbors that we just obtained in above step. Thereafter we will calculate the accuracy and find confusion matrix for each k value i.e 1,11 and 21.

**Analysis**

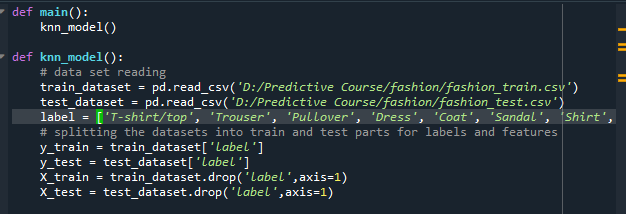
We have been provided with a “Fashion MNIST” data set and we have to apply classification on this image data set. So, I have used “sklearn” library available in python for implementing the “kNN algorithm”. Here is the logical analysis of my code:

Here are the important libraries that are required:



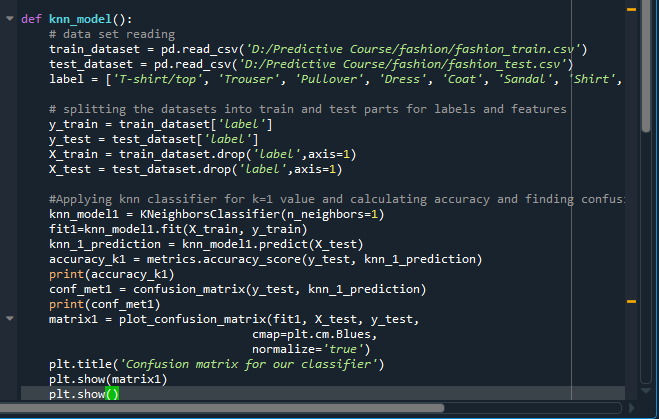
Then I created main() in which I am invoking the knn() where I wrote the logic for processing of dataset which is further feeded to build the model and then calculate the accuracy and confusion matrix.

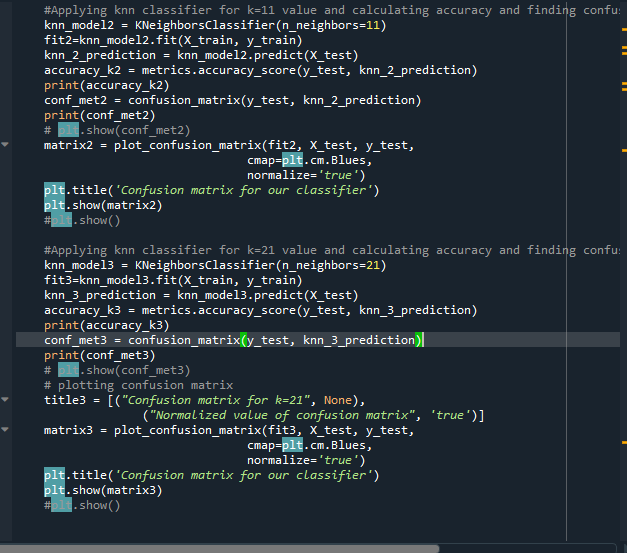
Since we have been provided with the train and test data sets so I splitted them into labels and features and passed the relevant values in the functions as follows:

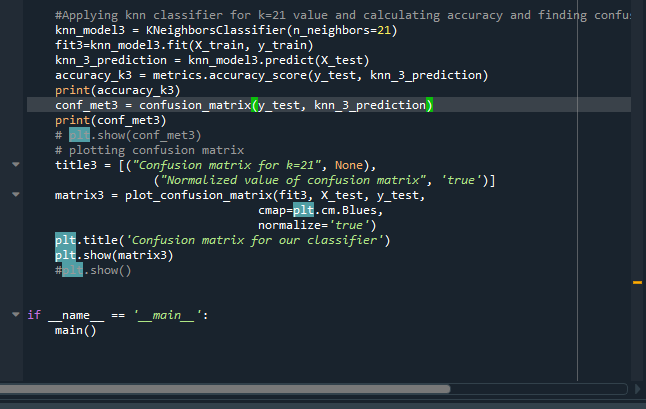


I have created classifiers for k=1,11 and 21 as mentioned in the question. And there after fitted the model for all the k values and generated confusion matrix for each and plotted them separately.

Here is the logic for the same:



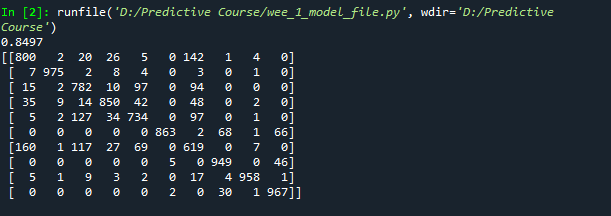


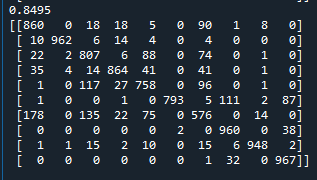


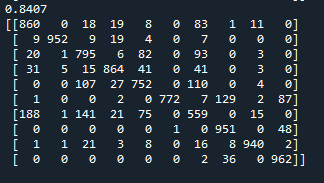
Accuracy score for k=1 is 84.97 which is pretty good value.

I have generated confusion matrix with 2 different approaches to match and compare the values obtained.

Results for the same are as follows:





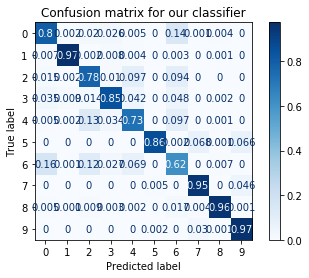


Similarly I have created separate variables for k=11 and k=21 to build “knn classifier” and finding out accuracy for each and then plotting the confusion matrix for them.

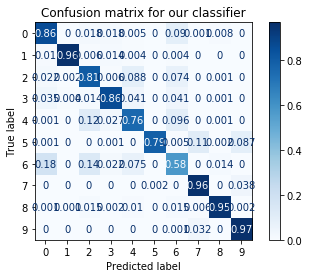
**Conclusion**

The results obtained for different k values i.e 1,11 and 21 showed a minimal difference in the accuracy i.e 0.02 % between k=1 and k=11, and for k=21 shows a little noticeable change but decrease in accuracy . As we can see that the accuracy for k=1 is 84.97 ,accuracy for k=11 is 84.95 and for k=21 is 84.07, which shows very minimal difference in the values resulting in conclusion that the image data set doesn’t show much difference if we increase the neighbors while applying “kNN algorithm” . The accuracy of this image data set can we increased by performing feature scaling or we can apply different algorithm. Plots for various k values are showing this trend. So we can conclude that the accuracy is decreasing if the value of k increases.

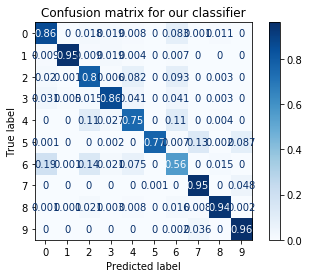
K = 1



K= 11



K =21



# References

1. Brownlee, J. (2020, February 23). Develop k-Nearest Neighbors in Python From Scratch. Retrieved September 22, 2020, from <https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/>
2. Radečić, D. (2020, January 06). Let's Make a KNN Classifier from Scratch. Retrieved September 22, 2020, from <https://towardsdatascience.com/lets-make-a-knn-classifier-from-scratch-e73c43da346d>
3. https://www.youtube.com/watch?v=44jq6ano5n0&list=PLQVvvaa0QuDfKTOs3Keq\_kaG2P55YRn5v&index=13
4. <https://www.youtube.com/watch?v=1i0zu9jHN6U&list=PLQVvvaa0QuDfKTOs3Keq_kaG2P55YRn5v&index=14>
5. K-Nearest Neighbor classification. (n.d.). Retrieved September 22, 2020, from <https://gurus.pyimagesearch.com/lesson-sample-k-nearest-neighbor-classification/>
6. K-nearest Neighbors (KNN) Classification Model. (n.d.). Retrieved September 22, 2020, from <https://www.ritchieng.com/machine-learning-k-nearest-neighbors-knn/>
7. K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint. (n.d.). Retrieved September 22, 2020, from <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
8. Minksminks 2, Bninopaulbninopaul 1, User1644018user1644018 73966 silver badges55 bronze badges, & Wagner CiprianoWagner Cipriano 65988 silver badges1111 bronze badges. (1965, June 01). How can I plot a confusion matrix? Retrieved September 22, 2020, from <https://stackoverflow.com/questions/35572000/how-can-i-plot-a-confusion-matrix>
9. Confusion matrix¶. (n.d.). Retrieved September 23, 2020, from https://scikit-learn.org/stable/auto\_examples/model\_selection/plot\_confusion\_matrix.html