**KNN CLASSIFIER**

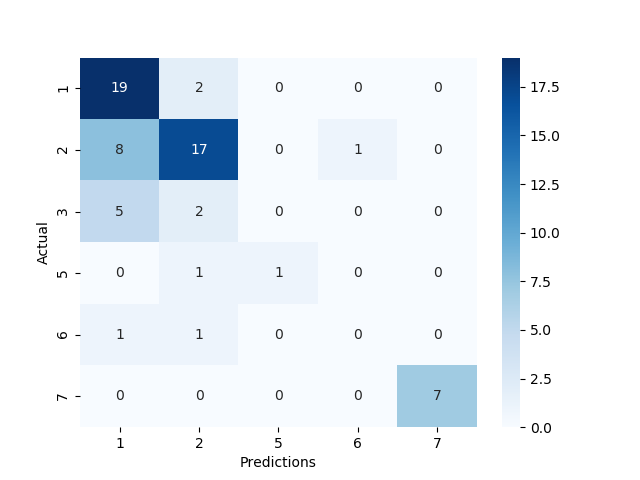
**Business Problem** = ﻿Prepare a model for glass classification using KNN.

* **Name of the File: -** glass.csv
* **Size of the File: -** 12 KB
* **Necessary Data : -** 215 Observations, 10 Features.

**Exploratory data Analysis** =

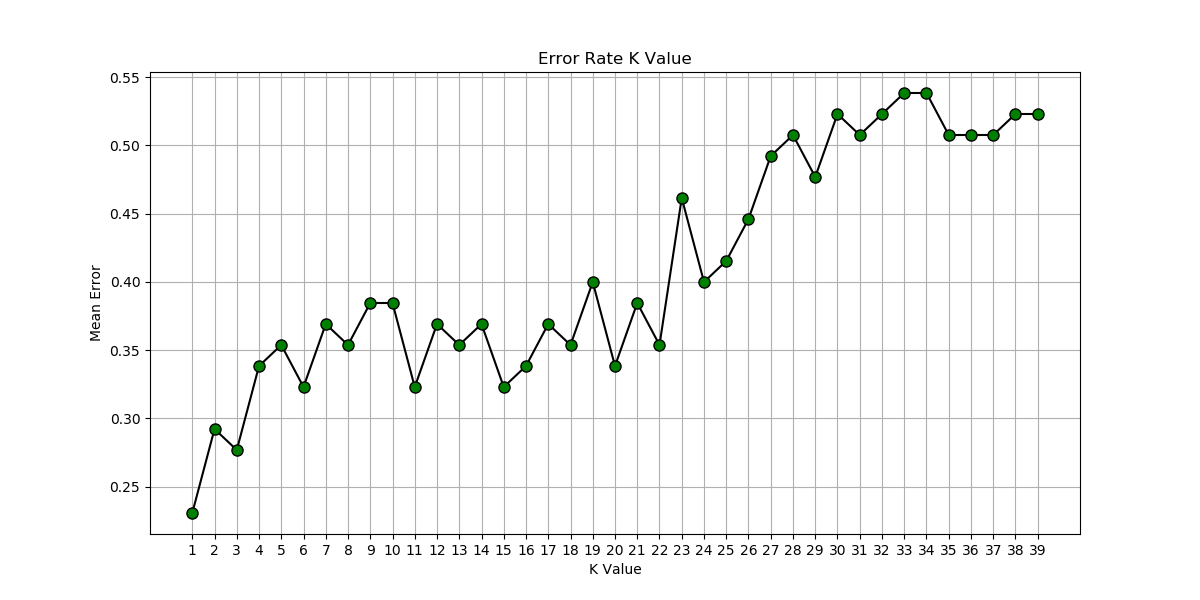
* **Outliers: -**  Outliers are not presents.
* **Missing Value: -** Data don’t have Missing Values
* **Output:** - Numeric

**Building Initial KNN Model =** Building the first KNN model by taking the square root of total observation in train data as a K value.

* **﻿N Neighbors (K) :-** 11
* **﻿Accuracy Score :-** 68 %
* **Confusion Matrix : -**

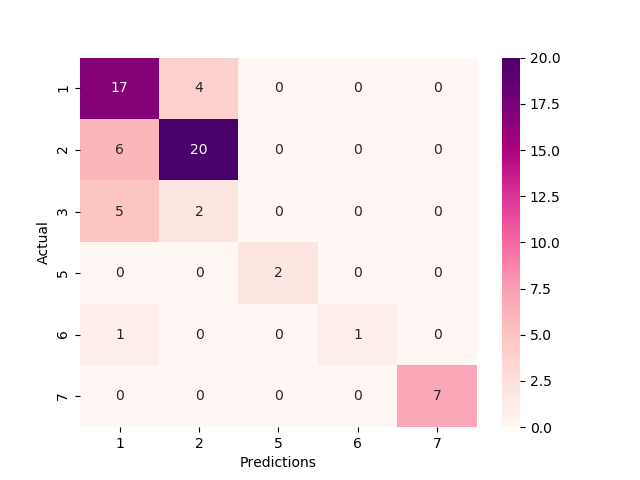
We can observe from the above confusion matrix and accuracy score the TP, as well as TN rate, is higher as compare to FP and FN but FP and FN rate is also quite high and we have to reduce it in order to improve model performance. Due to many rates of FP and FN, the accuracy of the model is also varies less it is near about 68%. We have to optimize our KNN model by selecting proper K value so that model gives high accuracy.

**Selecting Perfect K Value =** We can select optimized K value by performing the KNN model on different K values iteratively. I am using Elbow Curve to plot Absolute Error given by model on different K values.

* **﻿Error Elbow Plot : -**

From the above Elbow plot, we can conclude that the optimized odd K value for better accuracy and less Error is 3. Odd K value takes a decision based on the majority instead of distance Hence, it gives more accurate results. So we are training our final model by selecting K as 3.

**Building Final KNN Model =**

* **N Neighbors (K) :-** 3
* **﻿Accuracy Score :-** 72 %
* **Confusion Matrix : -**

After Selecting K as 3, our model accuracy apparently increased. Now model accuracy is greater than 72% and we can say that its quite good accuracy. So we use this model as our final model.

**Python code file**: - [Glass Analysis.py](https://github.com/nilaydeshmukh0/K-Nearest-Neighbor-Classifier-KNN/blob/master/Glass%20Analysis/Glass%20Analysis.py)