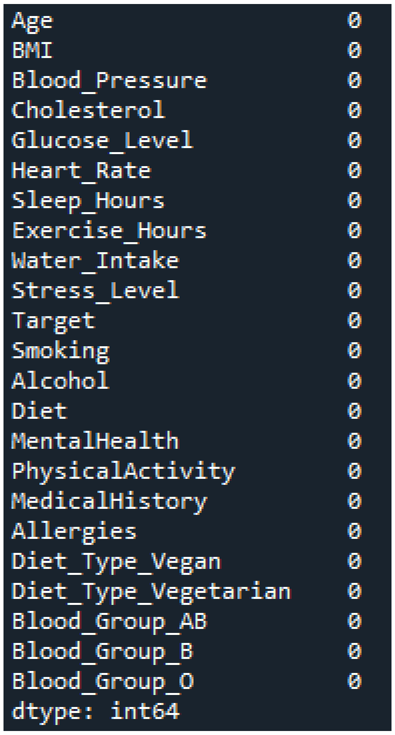
Problem Statement

Anova Insurance, a global health insurance company, seeks to optimize its insurance policy premium pricing based on the health status of applicants. Understanding an applicant's health condition is crucial for two key decisions:
- Determining eligibility for health insurance coverage.
- Deciding on premium rates, particularly if the applicant's health indicates higher risks.

The objective is to Develop a predictive model that utilizes health data to classify individuals as 'healthy' or 'unhealthy'. This classification will assist in making informed decisions about insurance policy premium pricing.

Analysis

After reading the dataset, we print the missing values in any column - there are no missing values -

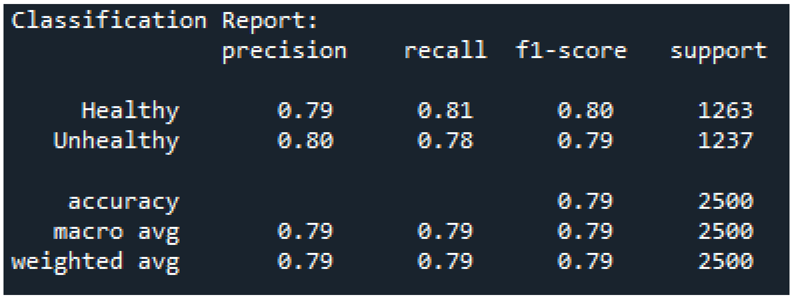


```
Age                    0
BMI                    0
Blood_Pressure         0
Cholesterol            0
Glucose_Level          0
Heart_Rate             0
Sleep_Hours            0
Exercise_Hours         0
Water_Intake           0
Stress_Level           0
Target                 0
Smoking                0
Alcohol                0
Diet                   0
MentalHealth           0
PhysicalActivity       0
MedicalHistory         0
Allergies              0
Diet_Type_Vegan        0
Diet_Type_Vegetarian   0
Blood_Group_AB         0
Blood_Group_B          0
Blood_Group_O          0
dtype: int64
```

We then convert the boolean data types in the diet and Blood group columns into integer form. After splitting the data into 75% train and 25% test data, a KNN model with k = 5 is chosen to fit the data.

The model accuracy score is calculated at **79.16%**, while a KFold cross validation shows a mean accuracy of **76.56%**.

The classification report shows the following statistics -

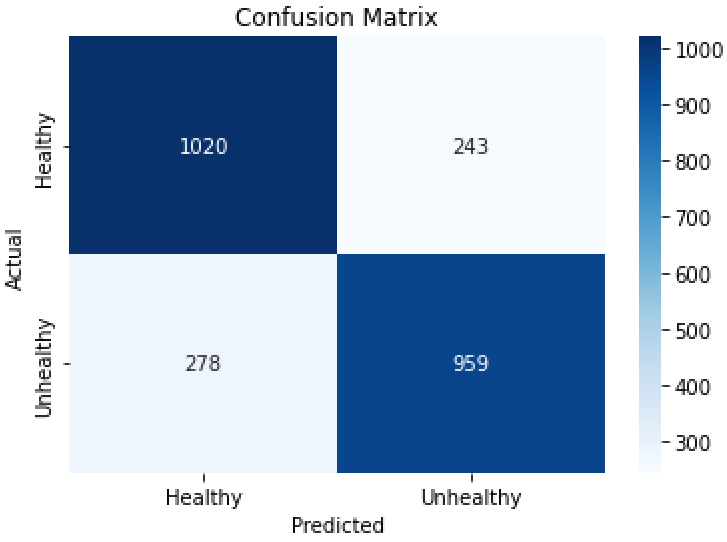


```
Classification Report:
               precision    recall  f1-score   support

     Healthy       0.79      0.81      0.80      1263
   Unhealthy       0.80      0.78      0.79      1237

    accuracy                           0.79      2500
   macro avg       0.79      0.79      0.79      2500
weighted avg       0.79      0.79      0.79      2500
```

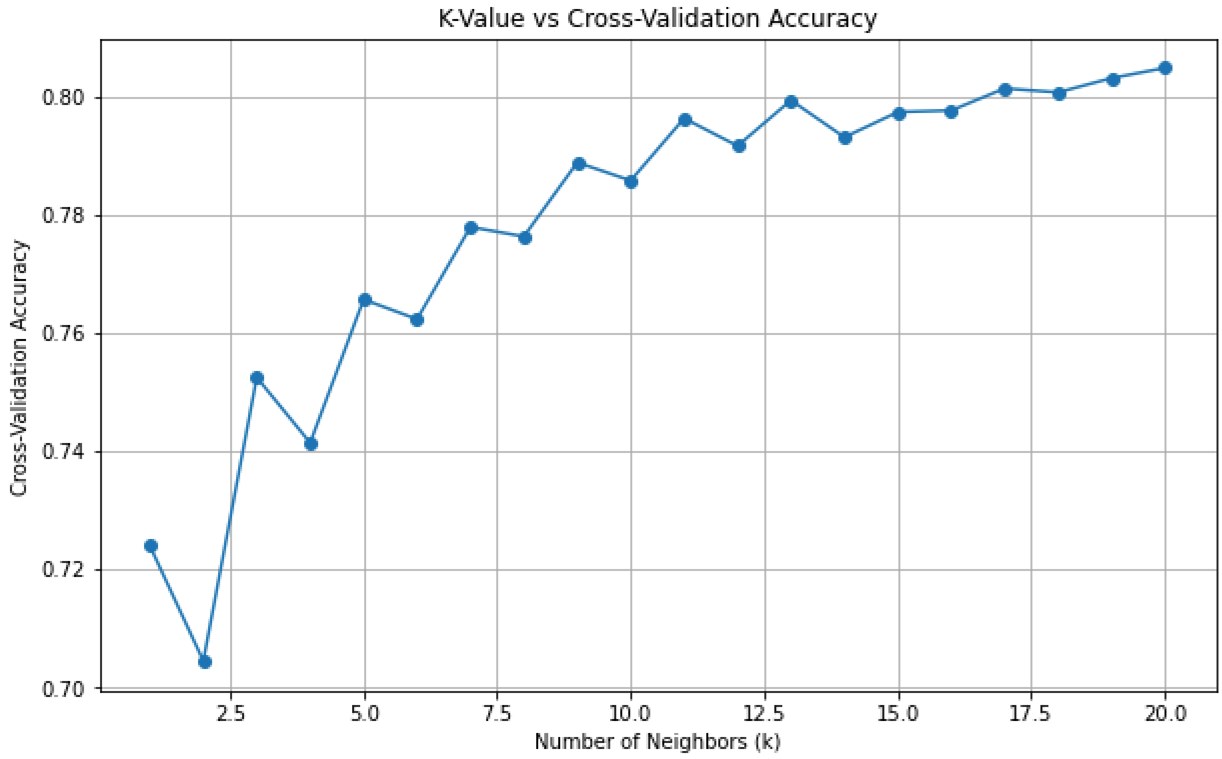
- **Precision:** Percentage of correct positive predictions relative to total positive predictions.
- **Recall:** Percentage of correct positive predictions relative to total actual positives.
- **F1 Score:** A harmonic mean of precision and recall. The closer to 1, the better the model.

A F1 score of 79% shows the model does fairly well in predicting the outcomes.

A confusion matrix



We also plot the cross-validation score for multiple values of k range from 1 to 20 to check for the best k value -

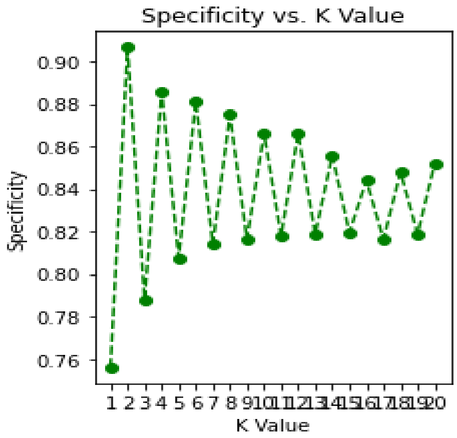


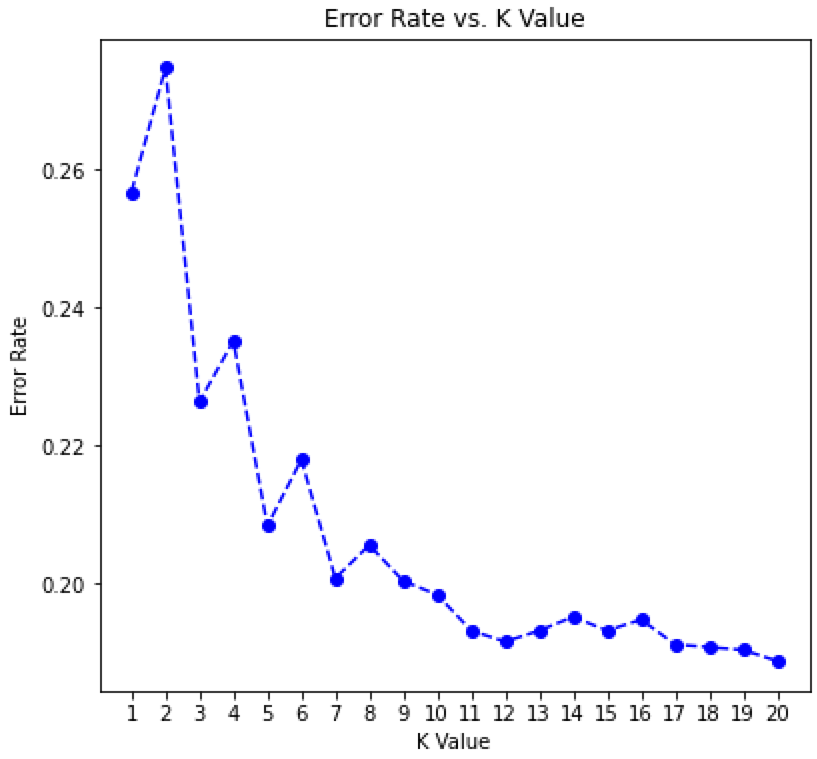
The above plot shows it reaches a high level of accuracy even at k = 5 (default) while slowly increasing the accuracy all the way up to k = 20 where the accuracy has increased from ~ 76% to 81%.

We can test the range of k-values also against the error rate and the specificity where the error rate and specificity are defined as-

- **Error Rate:** Proportion of incorrect predictions
- **Specificity**: True Negative Rate

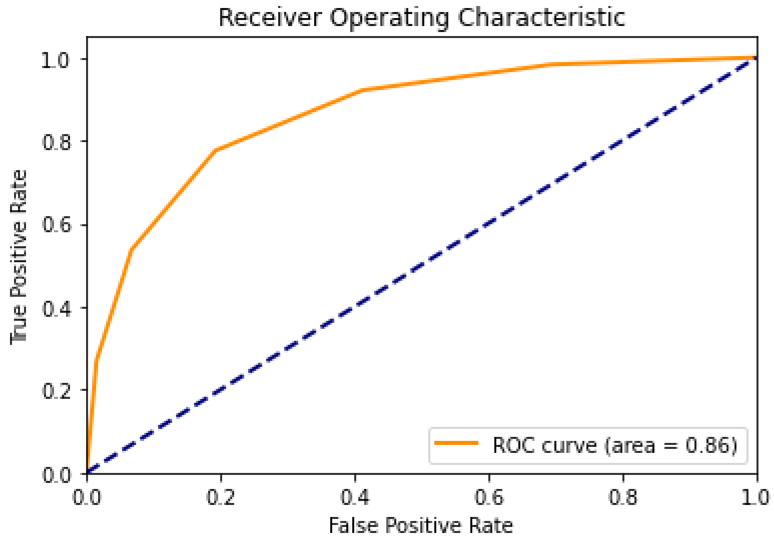
$$= True\ Negatives\ /\ (True\ Negatives\ +\ False\ Positives)$$



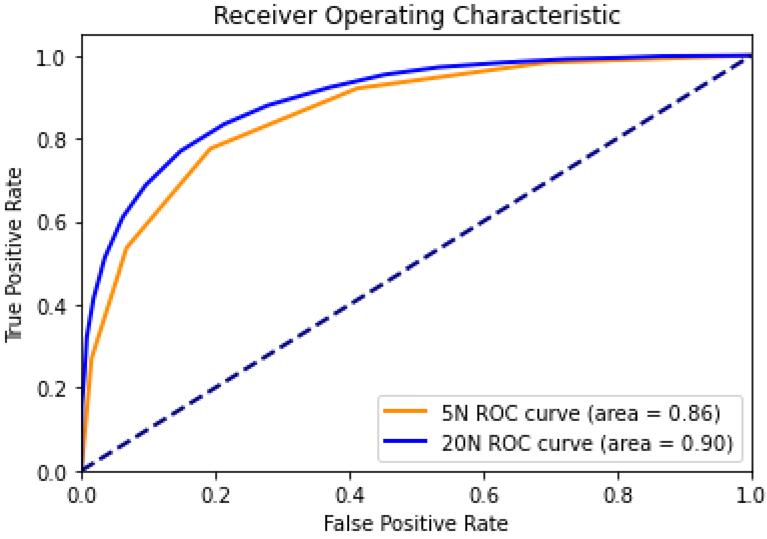


From the above plot, we again see that the error rate increases a lot by the time we have reached k = 5 and after k = 7 any additional increment in k-values doesn't massively help in dressing the error rate.

For the default KNN model with k = 5, we get the ROC curve where the area under curve is 86% -



Similarly, as a comparison between the different K values and plotting the ROC curve for K = 5 against k = 20, we see the ROC area under the curve only improving to 90% from 86%.



The above analysis shows we can fairly fit a KNN model with 5 knn-neighbors with a decent accuracy.