Cardiovascular Risk Prediction - Classification

- **Akash Choudhury**

Data science trainees,
AlmaBetter, Bangalore

Abstract:

Cardiovascular Diseases are the number one source of explanation for death globally, more people die annually from CVDs than from the other cause. Roughly estimated 17.9 million people died from CVDs in 2016, representing 31% of all global deaths. Early detection of cardiac diseases and continuous supervision of clinicians can reduce the death rate. Coming to CVD, the silver lining is that heart attacks are highly preventable and simple lifestyle modifications. It is, however, difficult to identify high risk patients because of the multifactorial nature of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, et cetera. This is where machine learning and data mining comes to the rescue. Our aim is to develop a machine learning model capable of identifying if a person has a cardiovascular disease or not. We can do this by extracting the features of a dataset and training a machine learning model with the highest accuracy. We trained four machine learning algorithms and compared their accuracy: Logistic Regression, K-nearest Neighbors ,Decision Tree Classifier and Support Vector Machine to identify people with high risk of getting cardiovascular disease.

Keywords: ***pandas, matplotlib, smote, model training, heart diesease, medical risk factor, approach.***

Problem Statement :-

Cardiovascular diseases (CVD) are currently the leading cause of premature death worldwide. The dataset is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD).

Introduction :-

Coronary artery disease (CAD) is the most common heart disease seen in today's population worldwide. Recent studies of the American Heart Association have shown that coronary artery diseases recorded 13% death in the USA in 2018 and worldwide in 2015. Amongst other diseases CAD found to be one of the most common causes of death, with the record of 15.6% of all results across the globe. Because this disease is associated with modifiable risk factors which indirectly related with lifestyle and intervention, timing of detection and diagnostic accuracy are especially relevant in clinical management of patients with CAD.

Over the past years, approaches making a significant impact in the detection and diagnosis of diseases that include machine learning (ML). In general, 'training' an algorithm with a control dataset for which the

disease status (disease or no disease) is known, and then applying this trained algorithm to a variable dataset in order to predict the disease status in patients for whom it is not yet determined. The ML algorithm will be better trained as a predictor for disease status as larger cohorts of data are introduced. prediction with ML would empower clinicians with improved detection, diagnosis, classification, risk stratification and ultimately, management of patients, all while potentially minimizing required clinical intervention are More accurate.

Data Description:-

Attribute Description :

Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors.

Demographic:

• Sex: male or female ("M" or "F")

• Age: Age of the patient;(Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)

Behavioral :

• is_smoking: whether or not the patient is a current smoker ("YES" or "NO")

• Cigs Per Day: the number of cigarettes that the person smoked on average in one day.(can be considered continuous as one can have any number of cigarettes, even half a cigarette.)

Medical(history)

• BP Meds: whether or not the patient was on blood pressure medication (Nominal)

• Prevalent Stroke: whether or not the patient had previously had a stroke (Nominal)

• Prevalent Hyp : whether or not the patient was hypertensive (Nominal)

• Diabetes: whether or not the patient had diabetes (Nominal)

Medical(current)

• Tot Chol: total cholesterol level (Continuous)

• Sys BP: systolic blood pressure (Continuous)

• Dia BP: diastolic blood pressure (Continuous)

• BMI: Body Mass Index (Continuous)

• Heart Rate: heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of a large number of possible values.)

• Glucose: glucose level (Continuous)

•Predict variable (desired target)

10-year risk of coronary heart disease CHD(binary: “1”, means “Yes”, “0” means “No”) - Dv

Challenges Faced :-

- As data is imbalanced, to balance the data in an appropriate way was a bit tricky.
- Using the appropriate metrics for comparison of the implemented machine learning algorithms.

Approach :-

We checked the Outliers and correlation matrix to overcome the noise in the dataset.

Also data was balanced using the SMOTE method and scaled by Standard Scaler transformation. As the Coronary Heart Diseases dataset defines the classification problem. We decided to train the models such as Logistic regression, K-nearest Neighbors , Decision Tree Classifier & Support Vector Machine .Also, we used Hyperparameter Tuning for improvement in the model fitting to understand the better results of the model as well as the metrics.

Data Cleaning :-

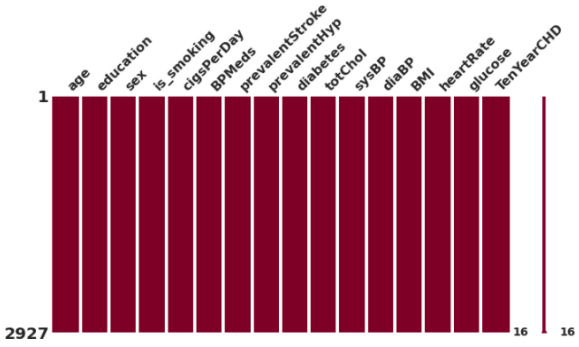
It is an essential skill of Data Scientists to be able to work with messy data, missing values, inconsistent, noise, or nonsensical data. To work smoothly python provides a built-in module Pandas

How do you remove NULL data in Python?

Pandas Data Frame drop na() function is used to remove rows and columns with Null/NaN values. By default, this function returns a new Data Frame and the source Data Frame remains unchanged. We can create null values using None, pandas. NaT, and numpy.

	name	age	marks
0	Joe	20	85.10
1	Sam	21	NaN
2	Harry	19	91.54

Pandas Data Frame drop na () function is used to remove rows and columns with Null/NaN values. By default, this function returns a new Data Frame and the source Data Frame remains unchanged We can create null values using None, pandas NaT, and numpy nan variables

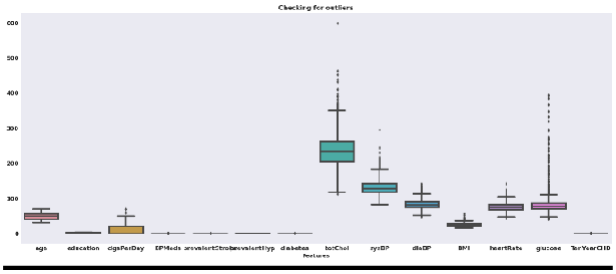


Visualization :-

Data Visualization represents the text or numerical data in a visual format, which makes it easy to grasp the information the data express. We, humans, remember the pictures more easily than readable text, so Python provides us various libraries for data visualization like matplotlib, seaborn, plotly, etc. In this tutorial, we will use Matplotlib and seaborn for performing various techniques to explore data using various plots. analysis is the simplest form of analysis where we explore a single variable. Univariate analysis is performed to describe the data in a better way. we perform Univariate analysis of Numerical and categorical variables differently because plotting uses different plots.

Outlier Detection :-

An outlier is a data point that is noticeably different from the rest. They represent errors in measurement, bad data collection, or simply show variables not considered when collecting the data.

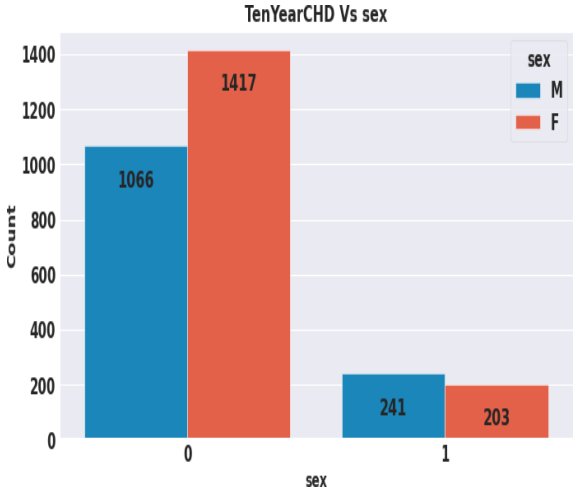


EDA

Bar Plot :-

Bar graphs/charts provide a visual presentation of categorical data.[4] Categorical data is a grouping of data into discrete groups, such as months of the year, age group, shoe sizes, and animals. These categories are usually qualitative. In a column (vertical) bar chart, categories appear along the horizontal axis and the height of the bar corresponds to the value of each category.

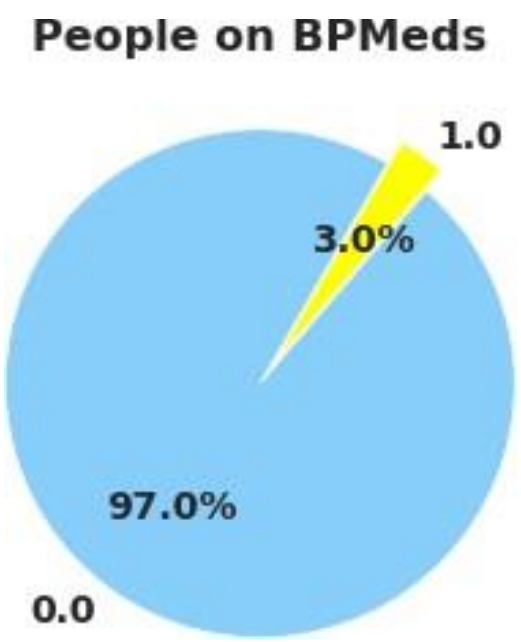
Bar charts have a discrete domain of categories, and are usually scaled so that all the data can fit on the chart. When there is no natural ordering of the categories being compared, bars on the chart may be arranged in any order. Bar charts arranged from highest to lowest incidence are called Pareto charts. A **bar chart** or **bar graph** is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column chart.



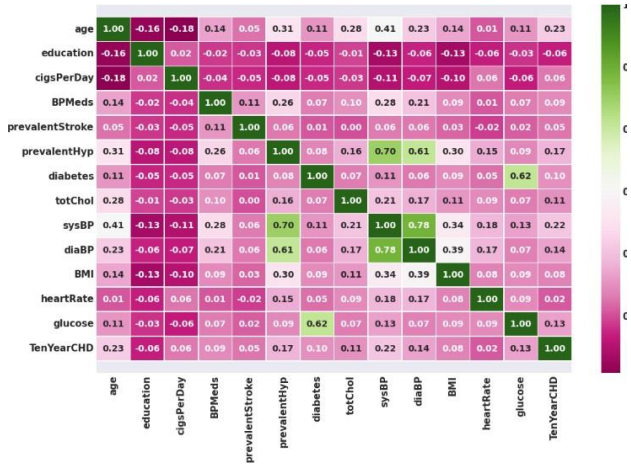
Pie Chart :-

A pie chart (or a circle chart) is a circular statistical graphic, which is divided into slices to illustrate numerical proportion. In a pie chart, the arc length of each slice (and consequently its central angle and area) is proportional to the quantity it represents. While it is named for its resemblance to a pie which has been sliced, there are variations on the way it can be presented. The earliest known pie chart is generally credited to William Playfair's *Statistical Breviary*

Pie charts are very widely used in the business world and the mass media.[3] However, they have been criticized,[4] and many experts recommend avoiding them,[5][6][7][8] as research has shown it is difficult to compare different sections of a given pie chart, or to compare data across different pie charts. Pie charts can be replaced in most cases by other plots such as the bar chart, box plot, dot plot, etc



CORRELATION MATRIX (HEATMAP)



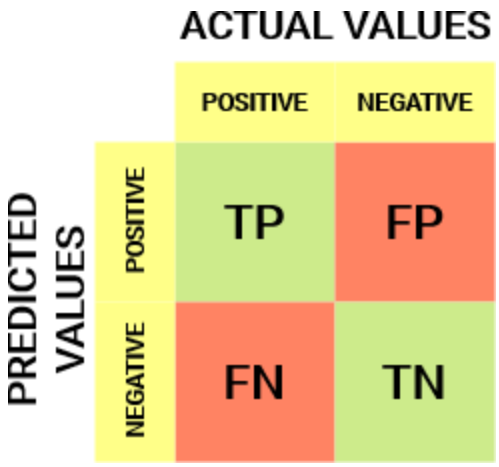
Seaborn

Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with data frames and the Pandas library. The graphs created can also be customized easily Data Visualization is the art of representing data in the form of graphs. It is a useful tool for professionals who work with data, i.e., financial analysts, business analysts, data analysts, data scientists This tutorial can be divided into three main parts. The first part will talk about installing seaborn and loading our dataset. In the second part, we will discuss some common graphs in Seaborn

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions. Seaborn is a Python data visualization library based on the Matplotlib library. It provides a high-level interface for drawing attractive and informative statistical graphs

Confusion Matrix

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.



True Positive (TP) -The predicted value matches the actual value. The actual value was positive and the model predicted a positive value.
True Negative (TN) -The predicted value matches the actual value.
The actual value was negative and the model predicted a negative value.
False Positive (FP) – Type 1 error :-The predicted value was falsely predicted. The actual value was negative but the model predicted a positive value.
False Negative (FN) – Type 2 error:-The predicted value was falsely predicted. The actual value was positive but the model predicted a negative value.

Logistic Regression

Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.
Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time. This implementation is for binary logistic regression. For data with more than 2 classes, softmax regression has to be used.

```
Train Result:
================================================
Accuracy Score: 85.94%

________________________________________________
CLASSIFICATION REPORT:
                    0            1  accuracy    macro avg  weighted avg
precision    0.860534     0.769231  0.859375     0.814882      0.847070
recall       0.996564     0.066225  0.859375     0.531394      0.859375
f1-score     0.923567     0.121951  0.859375     0.522759      0.805360
support   1746.000000   302.000000  0.859375  2048.000000   2048.000000
________________________________________________
Confusion Matrix:
 [[1740    6]
 [ 282   20]]

Test Result:
================================================
Accuracy Score: 84.87%

________________________________________________
CLASSIFICATION REPORT:
                    0           1  accuracy   macro avg  weighted avg
precision    0.848730    0.846154  0.848692    0.847442      0.848314
recall       0.997286    0.077465  0.848692    0.537376      0.848692
f1-score     0.917031    0.141935  0.848692    0.529483      0.791816
support    737.000000  142.000000  0.848692  879.000000    879.000000
________________________________________________
Confusion Matrix:
 [[735   2]
 [131  11]]
```

K-nearest Neighbors

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another. For classification problems, a class label is assigned on the basis of a majority vote—i.e. the label that is most frequently represented around a given data point is used. While this is technically considered “plurality voting”, the term, “majority vote” is more commonly used in literature. The distinction between these

terminologies is that “majority voting” technically requires a majority of greater than 50%, which primarily works when there are only two categories. Regression problems use a similar concept as classification problem, but in this case, the average the k nearest neighbors is taken to make a prediction about a classification. The main distinction here is that classification is used for discrete values, whereas regression is used with continuous ones. However, before a classification can be made, the distance must be defined. Euclidean distance is most commonly used, which we’ll delve into more below.

```
Train Result:
================================================
Accuracy Score: 86.67%
________________________________________________
CLASSIFICATION REPORT:
                    0           1  accuracy    macro avg  weighted avg
precision    0.876341    0.659341  0.866699     0.767841      0.844342
recall       0.982245    0.198675  0.866699     0.590460      0.866699
f1-score     0.926276    0.305344  0.866699     0.615810      0.834713
support    1746.000000  302.000000  0.866699  2048.000000   2048.000000
________________________________________________
Confusion Matrix:
 [[1715   31]
 [ 242   60]]

Test Result:
================================================
Accuracy Score: 82.82%
________________________________________________
CLASSIFICATION REPORT:
                    0           1  accuracy    macro avg  weighted avg
precision    0.845519    0.354839  0.828214     0.600179      0.766251
recall       0.972863    0.077465  0.828214     0.525164      0.828214
f1-score     0.904732    0.127168  0.828214     0.515950      0.779119
support    737.000000  142.000000  0.828214  879.000000    879.000000
________________________________________________
Confusion Matrix:
 [[717  20]
 [131  11]]
```

Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector. The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

```
Train Result:
================================================
Accuracy Score: 85.99%
________________________________________________
CLASSIFICATION REPORT:
                    0           1  accuracy    macro avg  weighted avg
precision    0.858829    1.000000  0.859863     0.929415      0.879646
recall       1.000000    0.049669  0.859863     0.524834      0.859863
f1-score     0.924054    0.094637  0.859863     0.509346      0.801747
support    1746.000000  302.000000  0.859863  2048.000000   2048.000000
________________________________________________
Confusion Matrix:
 [[1746    0]
 [ 287   15]]

Test Result:
================================================
Accuracy Score: 83.85%
________________________________________________
CLASSIFICATION REPORT:
                    0      1  accuracy    macro avg  weighted avg
precision    0.838453    0.0  0.838453     0.419226      0.703003
recall       1.000000    0.0  0.838453     0.500000      0.838453
f1-score     0.912129    0.0  0.838453     0.456064      0.764777
support    737.000000  142.0  0.838453  879.000000    879.000000
________________________________________________
Confusion Matrix:
 [[737   0]
 [142   0]]
```

Decision Tree Classifier

Decision tree build classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the

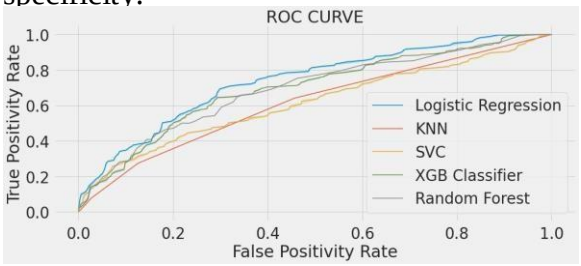
easiest and popular classification algorithms to understand and interpret. Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node

```
Train Result:
================================================
Accuracy Score: 100.00%
________________________________________________
CLASSIFICATION REPORT:
                0      1  accuracy  macro avg  weighted avg
precision     1.0    1.0       1.0        1.0           1.0
recall        1.0    1.0       1.0        1.0           1.0
f1-score      1.0    1.0       1.0        1.0           1.0
support    1746.0  302.0       1.0     2048.0        2048.0
________________________________________________
Confusion Matrix:
 [[1746    0]
 [   0  302]]

Test Result:
================================================
Accuracy Score: 77.36%
________________________________________________
CLASSIFICATION REPORT:
                    0           1  accuracy   macro avg  weighted avg
precision    0.862534    0.291971  0.773606    0.577252      0.770361
recall       0.868385    0.281690  0.773606    0.575038      0.773606
f1-score     0.865450    0.286738  0.773606    0.576094      0.771960
support    737.000000  142.000000  0.773606  879.000000    879.000000
________________________________________________
Confusion Matrix:
 [[640  97]
 [102  40]]
```

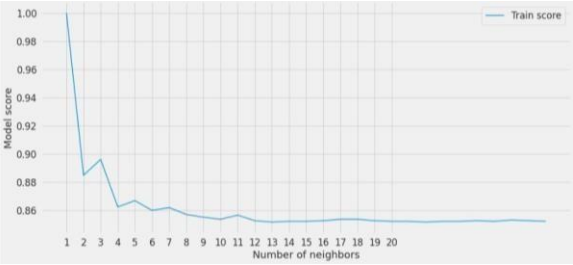
ROC Curve

This flexibility comes from the way that probabilities may be interpreted using different thresholds that allow the operator of the model to trade-off concerns in the errors made by the model, such as the number of false positives compared to the number of false negatives. This is required when using models where the cost of one error outweighs the cost of other types of errors. just know that the AUC-ROC curve helps us visualize how well our machine learning classifier is performing. Although it works for only binary classification problems, we will see towards the end how we can extend it to evaluate multi-class classification problems too. The paper by Body *et al* is concerned with the evaluation of decision aids, which can be used to identify potential acute coronary syndromes (ACS) in the ED. The authors previously developed the Manchester Acute Coronary Syndromes model (MACS) decision aid, which uses several clinical variables and two biomarkers to 'rule in' and 'rule out' ACS. However, one of the two biomarkers (heart-type fatty acid binding protein, H-FABP) is not widely used so a revised decision aid has been developed (Troponin-only Manchester Acute Coronary Syndromes, T-MACS), which include a single biomarker hs-cTnT. In this issue, the authors show how they derive a revised decision aid and describe its performance in a number of independent diagnostic cohort studies. Decision aids (as well as other types of 'diagnostic tests') are often evaluated in terms of diagnostic testing parameters such as the area under the receiver operating characteristic (ROC) curve, sensitivity and specificity. In this article, we explain how the ROC analysis is conducted and why it is an essential step towards developing a test with the desirable levels of sensitivity and specificity.



Hyperparameter Tuning on K-nearest Neighbors algorithm

hyperparameter tuning are called for. What distinguishes them is whether they come before (hyperparameter) or after (parameter) a model has been fit.KNN is a relatively simple classification tool, yet it's also highly effective much of the time. It gets bandied about that in approximately a third of all grouping cases**,** it's the most effective categorizer. A third! This model may be small, but so too is it mighty. There are different iterations of the algorithm in various programming languages, but I'll be discussing scikit-learn's here, which is in Python. The base algorithm uses Euclidean distance to find the nearest K (with K being our hyperparameter) training set vectors, or "neighbors," for each row in the test set. Majority vote decides what the classification will be, and if there happens to be a tie the decision goes to the neighbor that happened to be listed first in the training data.



Conclusion

	Model	Training Accuracy %	Testing Accuracy %
0	Logistic Regression	85.937500	84.869170
1	K-nearest neighbors	86.669922	82.821388
2	Support Vector Machine	85.986328	83.845279
3	Decision Tree Classifier	100.000000	77.360637

Logistic Regression, K Nearest Neighbors, Support Vector Machines, and Decision Tree Classification Models have been implemented. From these above models, we found ANN to be the best model compared to the other model

In Hyperparameter tuning ,we observed that K-Nearest Neighbors accuracy has improved which shows that KNN (with Hyperparameter Tuning) is the best fitted model for Coronary Heart Disease dataset.

Training Accuracy = 85.30 & Testing Accuracy = 84.07

We can also run random forest classifiers and XGBoost models for improved future coronary artery disease model fitting. By consulting the medical staff, the characteristics can be analyzed in an appropriate and necessary way to address the causes and consequences of the disease

	Model	Training Accuracy %	Testing Accuracy %
0	Tuned K-nearest neighbors	85.302734	84.07281

Reference

- https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5
- https://medium.datadriveninvestor.com/k-nearest-neighbors-in-python-hyperparameters-tuning-716734bc557f
- https://towardsdatascience.com/smote-fdce2f605729