# TITLE PAGE

**Submitted by**: Dhanavath Pavani

**Email.id**: [dhanavath.pavani.eee20@itbhu.ac.in](mailto:dhanavath.pavani.eee20@itbhu.ac.in)

**Institute**: Indian Institute of Technology Varanasi

**Company Name**: Smart Bridge

**Duration:** 14.06.2023 **-** 9.07.2023

**Under the guidance of :**

Project managers of Smart Bridge(Saumya Mohandas)

**Index page**

**1.Overview on Project**

**2.Data collection**

**3.Data preprocessing**

Importing the dataset.

* Data Cleaning
* Visualization
* EDA
* Analyse the data
* Taking care of Missing Data

**4.Model Building**

* + Import the model building Libraries
  + Initializing the model
  + Adding Machine Learning Model
  + SVM classifier
  + Random Forest classifier
  + Logistic Regression
  + Naïve Bayes classifier
  + Training the model
  + Model Evaluation
  + Save the Model

**5.Application Building Using Flask**

* Create an HTML file
* Build Python Code

**Liver Patient Analysis**

**1.Introduction to project:**

Liver diseases averts the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of liver disease at an early stage is a complex task for the doctors. The main objective of this project is to analyse the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This paper focuses on the related works of various authors on liver disease such that algorithms were implemented using Weka tool that is a machine learning software written in Java. Various attributes that are essential in the prediction of liver disease were examined and the dataset of liver patients were also evaluated. This paper compares various classification algorithms such as Random Forest, Logistic Regression and Separation Algorithm with an aim to identify the best technique. Based on this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver diseaserecommended to the user.

**2: Data Collection**

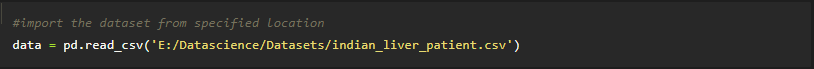
ML depends heavily on data, without data, a machine can't learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.

**The data wich I have used in model training is taken from Kaggle.**

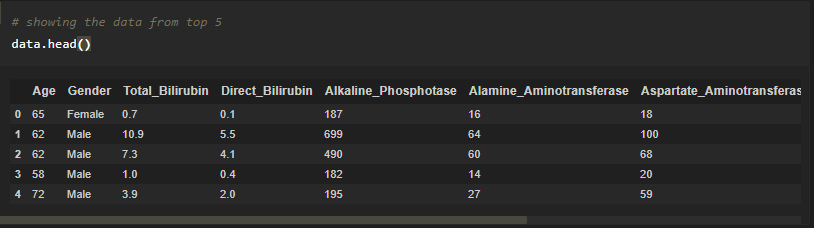
**3.Data preprocessing**

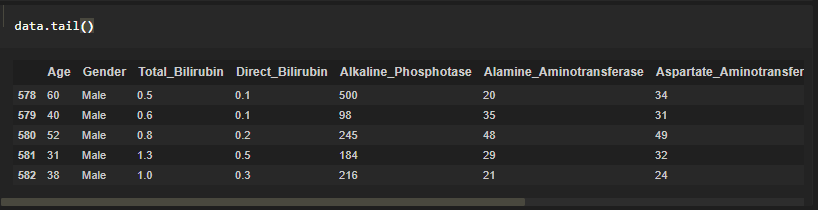
**Importing the dataset**

* You might have your data in .csv files, .excel files



Preview of the data set

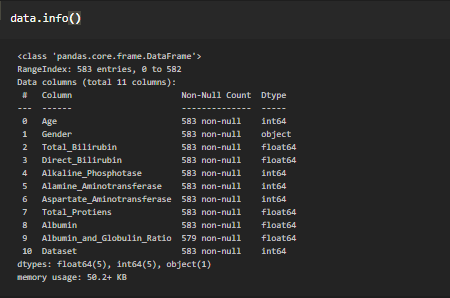




* describe() method computes a summary of statistics like count, mean, standard deviation, min, max, and quartile values.



info() gives information about the data



* **Hnadaling missing valuesing using mean and mode**

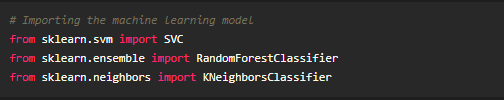
**Data Visualization**

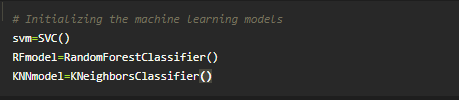


* Next step is trai test split

**Model Building**

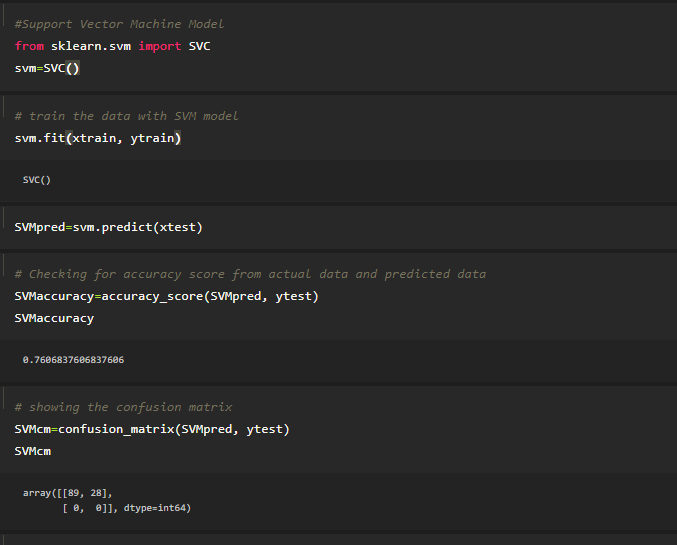
Importing the necessary libraries

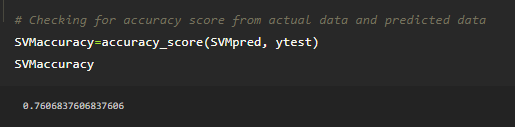




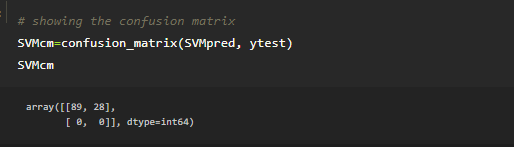
**Support Vector Machine**

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. Support Vectors are simply the co-ordinates of individual observation. The goal of a support vector machine is not only to draw hyperplanes and divide data points, but to draw the hyperplane the separates data points with the largest margin, or with the most space between the dividing line and any given data point



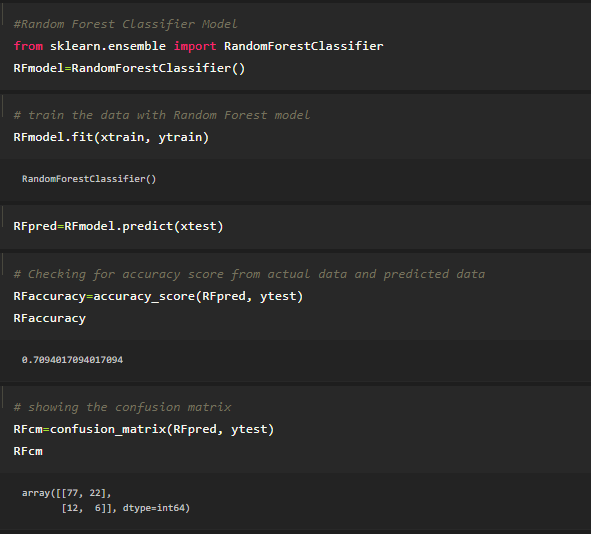


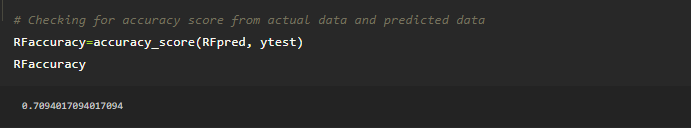
**Confusion\_matrix():**  A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. The rows represent the predicted values of the target variable.



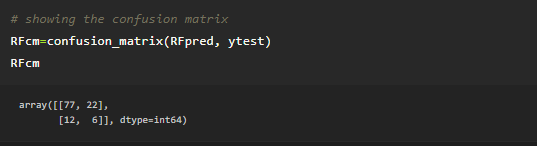
**Random Forest Algorithm**

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.



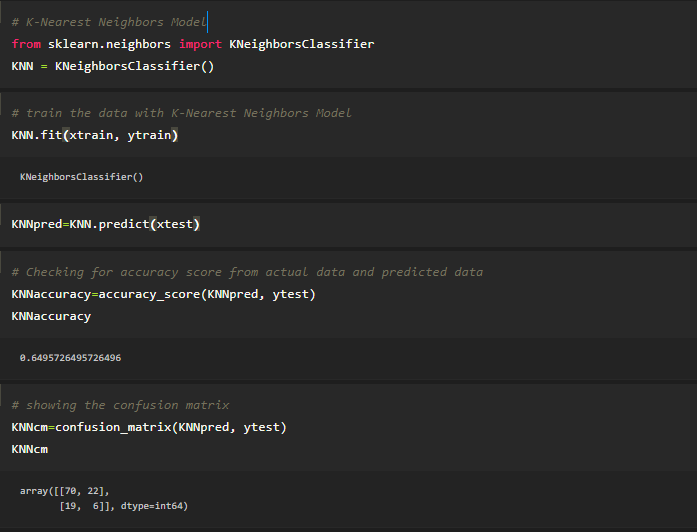
****

**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. The rows represent the predicted values of the target variable.

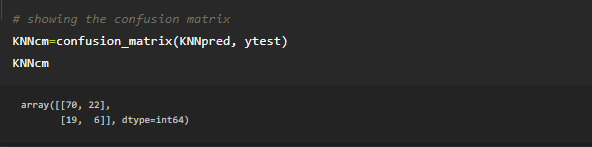


**K-Nearest Neighbors** **Algorithm**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

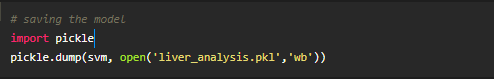


**Confusion\_matrix():**  A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. The rows represent the predicted values of the target variable.



**Save the Model**

The model is saved with .pkl extension as follows

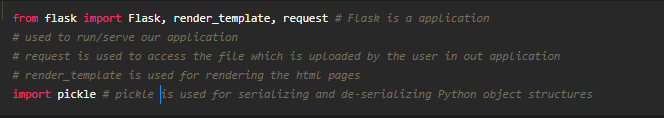


**Model application**

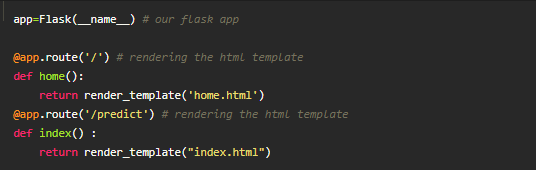
**Build python code**

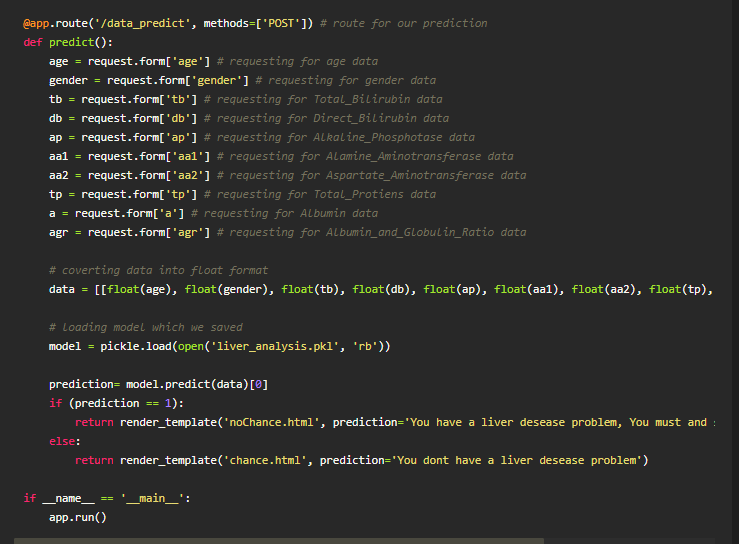
* Let us build flask file ‘Liver\_Flask\_App.ipynb’ which is a web framework written in python for server-side scripting. Let’s see step by step procedure for building the backend application.
* App starts running when “\_\_name\_\_” constructor is called in main.
* render\_template is used to return html file.
* “GET” method is used to take input from the user.
* “POST” method is used to display the output to the user.

* Importing Libraries

****

* Routing to the html Page





**Activity 3: Run The app in local browser**

* Open jupyter notebook from the start menu
* Navigate to the folder where your python script is.
* Now press Ctrl+Enter
* Navigate to the localhost where you can view your web page

