TEAM ID	PNT2022TMID42133
PROJECT NAME	Statistical Machine Learning Approaches to Liver Disease Prediction
COLLEGE NAME	AVS College Of Technology

Team Leader : T.Ramesh

Team Member: G.Karunakaran

Team Member: A.Deepak

Team Member: P.Hariprasath

Train and Test the Model Using Classification Algorithms

There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have may be Classification algorithms are Regression algorithms.

Example:

1. Random Forest Classification.

2. Support Vector Machine

3. KNN Classification

You will need to train the datasets to run smoothly and see an incremental improvement in the prediction rate.

Now we apply classification algorithms on our dataset.

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.

Random Forest Regression: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean/average prediction of the individual trees.

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.

Now we apply classification algorithms on our dataset.

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.

Random Forest Regression: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean/average prediction of the individual trees.

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.

Build the model

We're going to use x_train and y_train obtained above in train_test_split section to train our regression model. We're using the fit method and passing the parameters as shown below.

1. Import the Classification algorithms

```
# Importing the machine learning model
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

2. Initialize the model

```
# Initializing the machine learning models
svm=SVC()
RFmodel=RandomForestClassifier()
KNNmodel=KNeighborsClassifier()
```

3. Training model with our data.

- SVC Model

```
#Support Vector Machine Model
from sklearn.svm import SVC
svm=SVC()

# train the data with SVM model
svm.fit(xtrain, ytrain)

 SVC()
```

Random Forest Model

```
#Random Forest Classifier Model
from sklearn.ensemble import RandomForestClassifier
RFmodel=RandomForestClassifier()

# train the data with Random Forest model
RFmodel.fit(xtrain, ytrain)

  RandomForestClassifier()
```

Training and Testing the Model :

Train and Test The Model using Classification Algorithm

In [1]:
```
import numpy as np
import pandas as pd
from sklearn import ensemble
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import joblib
```

In [2]:
```
patients=pd.read_csv(r'C:\Users\WELCOME\AppData\Local\Programs\Python\Python310\Lib\MyProject\indian_liver_patient.csv')
patients['Gender']=patients['Gender'].apply(lambda x:1 if x=='Male' else 0)
patients=patients.fillna(0.94)
```

In [3]:
```
X=patients[['Total_Bilirubin', 'Direct_Bilirubin',
       'Alkaline_Phosphotase', 'Alamine_Aminotransferase',
       'Total_Protiens', 'Albumin', 'Albumin_and_Globulin_Ratio']]
y=patients['Dataset']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=123)

print('Shape training set: X:{}, y:{}'.format(X_train.shape, y_train.shape))
print('Shape test set: X:{}, y:{}'.format(X_test.shape, y_test.shape))
```

```
Shape training set: X:(6703, 7), y:(6703,)
Shape test set: X:(2874, 7), y:(2874,)
```

In [4]:
```
model = ensemble.RandomForestClassifier()
model.fit(X_train, y_train) #Put X_Train.values while running The app.py---
y_pred = model.predict(X_test)
print('Accuracy : {}'.format(accuracy_score(y_test, y_pred)))

clf_report = classification_report(y_test, y_pred)
print('Classification report')
print("---------------------")
print(clf_report)
print("_____________________")
```

```
Accuracy : 1.0
Classification report
---------------------
              precision    recall  f1-score   support

           1       1.00      1.00      1.00      2038
           2       1.00      1.00      1.00       836

    accuracy                           1.00      2874
   macro avg       1.00      1.00      1.00      2874
weighted avg       1.00      1.00      1.00      2874

_____________________
```