**DIABETES**

**Model Used:**

**Random Forest:**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

"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. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

**SVM:**

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.

**Model Building**

**Splitting the dataset**

X = diabetes\_df.drop('Outcome', axis=1)

y = diabetes\_df['Outcome']

**Now we will split the data into training and testing data using the train\_test\_split function**

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size=0.33, random\_state=7)

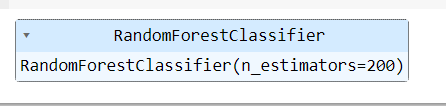
**Random Forest**

**Building the model using RandomForest**

from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(n\_estimators=200)

rfc.fit(X\_train, y\_train)



**Now after building the model let’s check the accuracy of the model on the training dataset.**

rfc\_train = rfc.predict(X\_train)

from sklearn import metrics

print("Accuracy\_Score =", format(metrics.accuracy\_score(y\_train, rfc\_train)))



**Getting the accuracy score for Random Forest**

from sklearn import metrics

predictions = rfc.predict(X\_test)

print("Accuracy\_Score =", format(metrics.accuracy\_score(y\_test, predictions)))

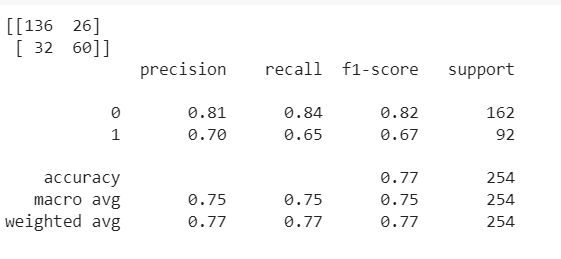


**Classification report and confusion matrix of random forest model**

from sklearn.metrics import classification\_report, confusion\_matrix

print(confusion\_matrix(y\_test, predictions))

print(classification\_report(y\_test,predictions))



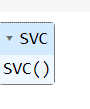
## Support Vector Machine (SVM)

**Building the model using Support Vector Machine (SVM)**

from sklearn.svm import SVC

svc\_model = SVC ()

svc\_model.fit(X\_train, y\_train)



**Prediction from support vector machine model on the testing data**

svc\_pred = svc\_model.predict(X\_test)

**Accuracy score for SVM**

from sklearn import metrics

print("Accuracy Score =", format(metrics.accuracy\_score(y\_test, svc\_pred)))

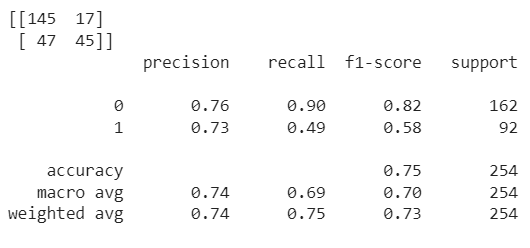


**Classification report and confusion matrix of the SVM classifier**

from sklearn.metrics import classification\_report, confusion\_matrix

print(confusion\_matrix(y\_test, svc\_pred))

print(classification\_report(y\_test,svc\_pred))



**Conclusion:**

After using all these patient records, we are able to build a machine learning model (random forest – best one) to accurately predict whether or not the patients in the dataset have diabetes or not along with that we were able to draw some insights from the data via data analysis and visualization.