# Topic(s): Decision Tree & Random Forest

**Problem Statement: -**

 A cloth manufacturing company is interested to know about the segment or attributes contributing to high sale. Approach - A decision tree & random forest model can be built with target variable 'Sales' (we will first convert it into categorical variable) & all other variables will be independent in the analysis.

Business objective:

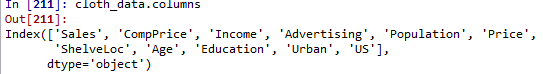
To classify sales of cloth manufacturing company by Decision tree and Random forest model .

Solution:

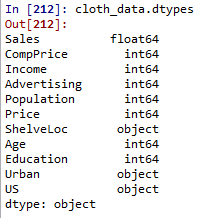
Load the company\_data.csv data file which contains all the data about clothing industry .

Exploring Data

After you have loaded the dataset, you might want to know a little bit more about it. You can check feature and target names.



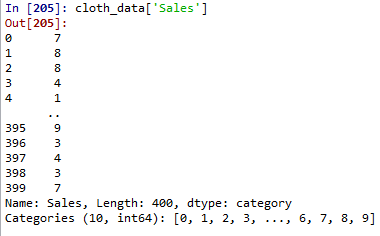
inspection of data types of each column



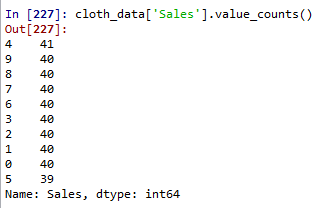
#converting categorical columns to numeric by using label encoder . Label Encoder is obtained importing from sklearn.preprocessing

# converting target values to category as we are using decision tree classification which doesn't works on continuous data.

cloth\_data['Sales']=cloth\_data['Sales'].astype("category")



By using unique( ) and values\_count() methods all unique entries and their count are measured in target data



#### Splitting Data

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

Let's split dataset by using function train\_test\_split(). You need to pass 3 parameters features, target, and test\_set size. Additionally, you can use random\_state to select records randomly.

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

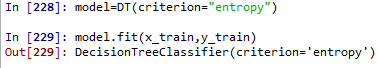
x\_train,x\_test,y\_train,y\_test=train\_test\_split(predictors,target,test\_size=0.2)

## Decision tree classifier model

The decision tree algorithm is build by calling of Decision tree classifier from sklearn.tree

from sklearn.tree import DecisionTreeClassifier as DT

the built model is fitted to train data in order to train the model



Let's estimate, how accurately the classifier or model can predict the type of cultivars.

Accuracy can be computed by comparing actual test set values and predicted values.



Accuracy score is measured



Well, you got a classification rate of 0.1, considered as low accuracy.

Random Forest Classifier model

The Random Forest Classifier model is an ensemble model where base learner is decisition tree algorithm

from sklearn.ensemble import RandomForestClassifier

the built model is fitted to train data in order to train the model

1)n\_estimators=15



Let's estimate, how accurately the classifier or model can predict the type.

Accuracy can be computed by comparing actual test set values and predicted values.



Accuracy score is measured

Score here is very low

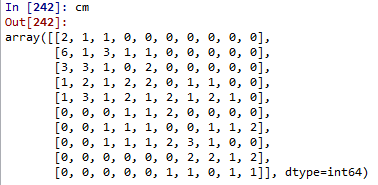
changing the value of n\_estimators which represents the no. of decision tree models .

n\_estimators=20

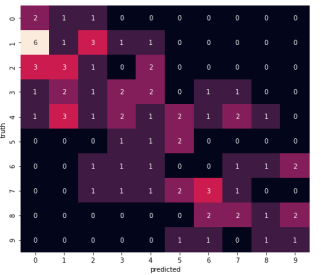


In this case also the accuracy is very low

to visualize the measure of accuracy confusion matrix is plotted



using it is better visualized



# Topic(s): Decision Tree & Random Forest

**Problem Statement: -**

 **Problem Statement: -**

Divide the data (Diabetes) into training and test datasets and create a Random Forest and Decision Tree Model to classify 'Class Variable' or “Outcome”

Business objective:

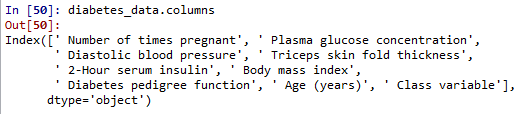
To classify class variable of diabetes data set by Decision tree and Random forest model .

Solution:

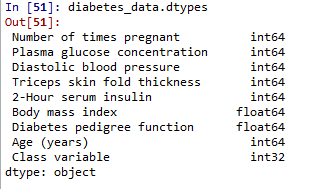
Load the Diabetes.csv data file which contains all the data about .

Exploring Data

After you have loaded the dataset, you might want to know a little bit more about it. You can check feature and target names.



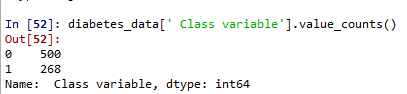
inspection of data types of each column



#converting categorical columns to numeric by using label encoder . Label Encoder is obtained importing from sklearn.preprocessing

converting class variable column to numeric by label encoder

By using unique( ) and values\_count() methods all unique entries and their count are measured in target data



#### Splitting Data

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

Let's split dataset by using function train\_test\_split(). You need to pass 3 parameters features, target, and test\_set size. Additionally, you can use random\_state to select records randomly.

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

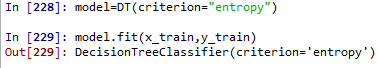
x\_train,x\_test,y\_train,y\_test=train\_test\_split(predictors,target,test\_size=0.2)

## Decision tree classifier model

The decision tree algorithm is build by calling of Decision tree classifier from sklearn.tree

from sklearn.tree import DecisionTreeClassifier as DT

the built model is fitted to train data in order to train the model



Let's estimate, how accurately the classifier or model can predict the type of cultivars.

Accuracy can be computed by comparing actual test set values and predicted values.



Accuracy score is measured



Well, you got a classification rate of 70.77% considered as good accuracy.

Random Forest Classifier model

The Random Forest Classifier model is an ensemble model where base learner is decisition tree algorithm

from sklearn.ensemble import RandomForestClassifier

the built model is fitted to train data in order to train the model

1)n\_estimators=15



Let's estimate, how accurately the classifier or model can predict the type.

Accuracy can be computed by comparing actual test set values and predicted values.



Accuracy score is measured

Score here is good

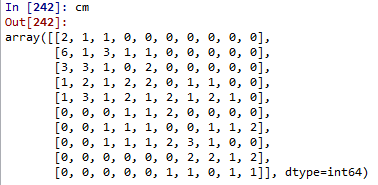
changing the value of n\_estimators which represents the no. of decision tree models .

n\_estimators=20

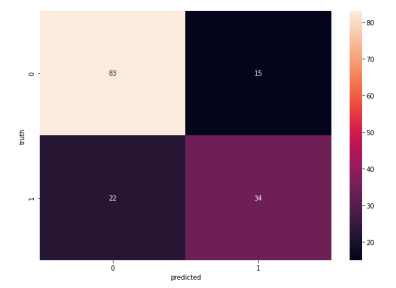


In this case also the accuracy is good

to visualize the measure of accuracy confusion matrix is plotted



using it is better visualized



Problem Statement: -

Use decision trees & random forest algorithm to prepare a model on fraud datatreating those who have taxable\_income <= 30000 as "Risky" and others are "Good".

Business objective:

To classify risky and good customers by Decision tree and Random forest model .

Solution:

Load the fraud\_check.csv data file which contains all the data about clothing industry .

Exploring Data

After you have loaded the dataset, you might want to know a little bit more about it. You can check feature and target names

inspection of data types of each column

#converting categorical columns to numeric by using label encoder . Label Encoder is obtained importing from sklearn.preprocessing

By using unique( ) and values\_count() methods all unique entries and their count are measured in target data

#### Splitting Data

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

Let's split dataset by using function train\_test\_split(). You need to pass 3 parameters features, target, and test\_set size. Additionally, you can use random\_state to select records randomly.

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

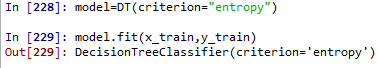
x\_train,x\_test,y\_train,y\_test=train\_test\_split(predictors,target,test\_size=0.2)

## Decision tree classifier model

The decision tree algorithm is build by calling of Decision tree classifier from sklearn.tree

from sklearn.tree import DecisionTreeClassifier as DT

the built model is fitted to train data in order to train the model



Let's estimate, how accurately the classifier or model can predict the type of cultivars.

Accuracy can be computed by comparing actual test set values and predicted values.

accuracy score is 74.67% which considered as good

Accuracy score is measured

Random Forest Classifier model

The Random Forest Classifier model is an ensemble model where base learner is decisition tree algorithm

from sklearn.ensemble import RandomForestClassifier

the built model is fitted to train data in order to train the model

1)n\_estimators=15



Let's estimate, how accurately the classifier or model can predict the type.

Accuracy can be computed by comparing actual test set values and predicted values.

Accuracy score is measured as 79%

Score here is good

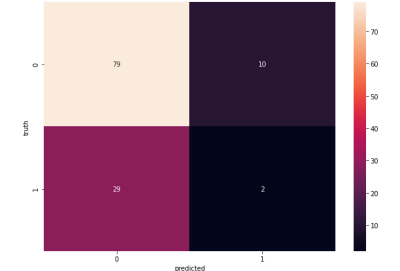
changing the value of n\_estimators which represents the no. of decision tree models .

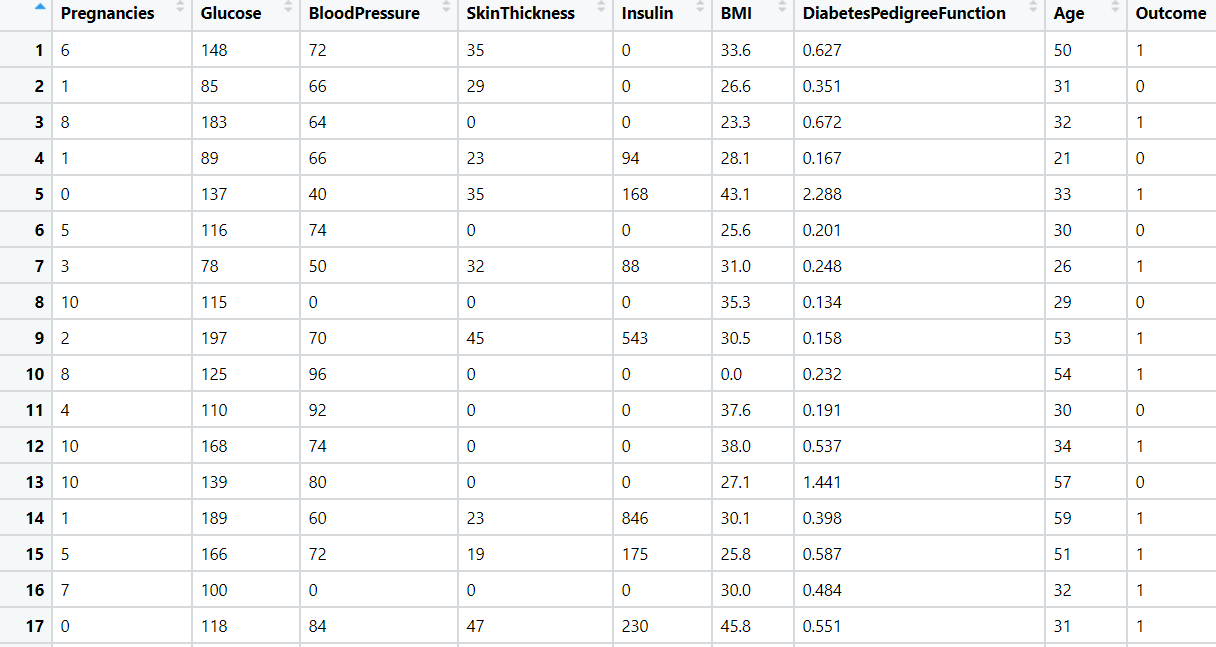
n\_estimators=20

In this case also the accuracy is very low

to visualize the measure of accuracy confusion matrix is plotted

using it is better visualized



Problem Statement: -

In Recruitment domain, HR faces with the challenge of predicting if the candidate is faking his salary or the candidate is genuine. In order to do it manually, let us use our Machine Learning algorithm to correctly classify using Decision Tree and Random Forest. We have a scenario where, a candidate claims to have 5 years of experience and earning 70000 per month working as regional manager and the candidate is expecting more than his previous CTC. A sample data has been collected, find out the candidate claims are genuine or fake.

Business objective:

To classify genuine and fake candidates by Decision tree and Random forest model .

Solution:

Load the fraud\_check.csv data file which contains all the data about clothing industry .

Exploring Data

After you have loaded the dataset, you might want to know a little bit more about it. You can check feature and target names

inspection of data types of each column

#converting categorical columns to numeric by using label encoder . Label Encoder is obtained importing from sklearn.preprocessing

By using unique( ) and values\_count() methods all unique entries and their count are measured in target data

#### Splitting Data

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

Let's split dataset by using function train\_test\_split(). You need to pass 3 parameters features, target, and test\_set size. Additionally, you can use random\_state to select records randomly.

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

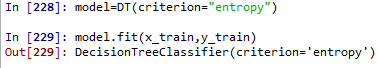
x\_train,x\_test,y\_train,y\_test=train\_test\_split(predictors,target,test\_size=0.2)

## Decision tree classifier model

The decision tree algorithm is build by calling of Decision tree classifier from sklearn.tree

from sklearn.tree import DecisionTreeClassifier as DT

the built model is fitted to train data in order to train the model



Let's estimate, how accurately the classifier or model can predict the type of cultivars.

Accuracy can be computed by comparing actual test set values and predicted values.

accuracy score is ver low

Accuracy score is measured

Random Forest Classifier model

The Random Forest Classifier model is an ensemble model where base learner is decisition tree algorithm

from sklearn.ensemble import RandomForestClassifier

the built model is fitted to train data in order to train the model

1)n\_estimators=15



Let's estimate, how accurately the classifier or model can predict the type.

Accuracy can be computed by comparing actual test set values and predicted values.

Accuracy score is measured as very low

Score here is good

changing the value of n\_estimators which represents the no. of decision tree models .

n\_estimators=20

In this case also the accuracy is very low

to visualize the measure of accuracy confusion matrix is plotted

using it is better visualized

