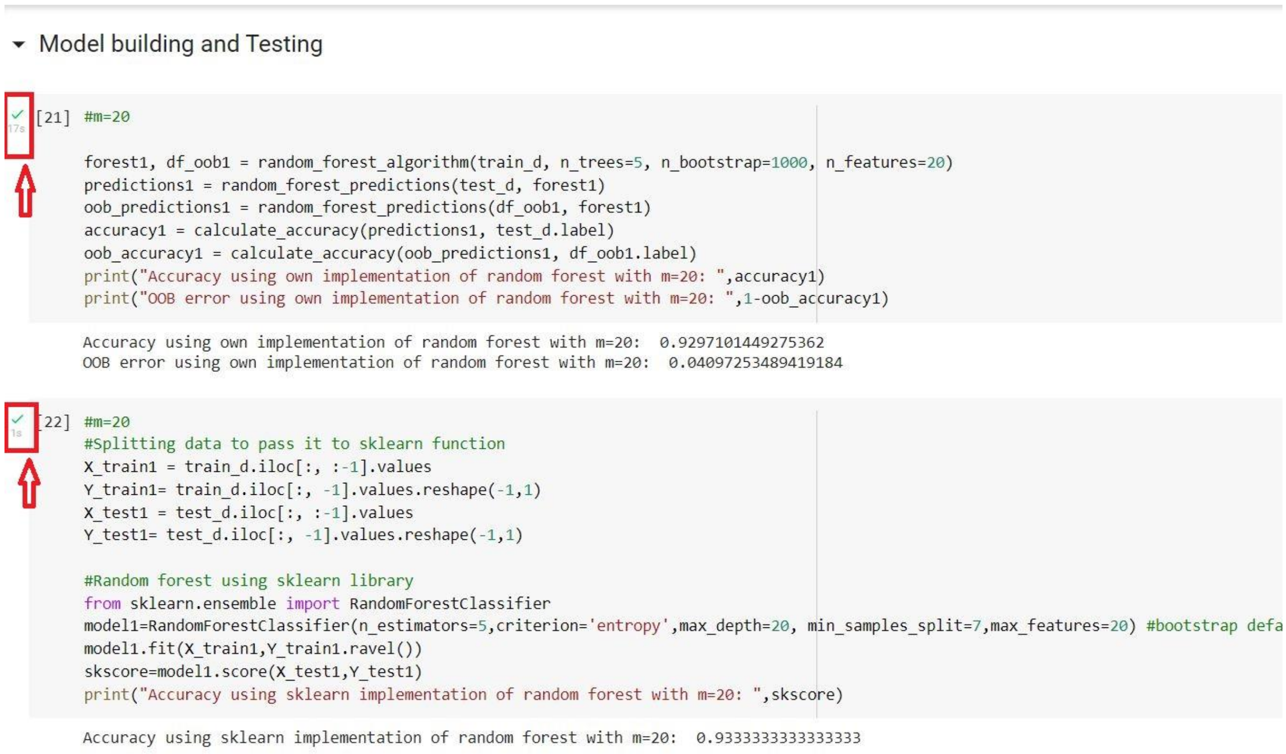
Qn.4a) Comparison of results in terms of accuracy and time taken between own random forest implementation and sklearn random forest implementation.

Note: The time taken for execution is obtained from running cells in google colab

No. of features (m)	Accuracy of own implementation of random forest	Accuracy of built in sklearn random forest	Time taken for own implementation of random forest(seconds)	Time taken for built in sklearn random forest (seconds)
20	0.9231884057971015	0.9355072463768116	17	1
25	0.9173913043478261	0.9282608695652174	20	1
30	0.9181159420289855	0.922463768115942	25	1
35	0.9195652173913044	0.9326086956521739	27	1
40	0.9239130434782609	0.922463768115942	31	1

Proof for time taken:



Model building and Testing

```
#m=20

forest1, df_oob1 = random_forest_algorithm(train_d, n_trees=5, n_bootstrap=1000, n_features=20)
predictions1 = random_forest_predictions(test_d, forest1)
oob_predictions1 = random_forest_predictions(df_oob1, forest1)
accuracy1 = calculate_accuracy(predictions1, test_d.label)
oob_accuracy1 = calculate_accuracy(oob_predictions1, df_oob1.label)
print("Accuracy using own implementation of random forest with m=20: ",accuracy1)
print("OOB error using own implementation of random forest with m=20: ",1-oob_accuracy1)
```

```
Accuracy using own implementation of random forest with m=20:  0.9297101449275362
OOB error using own implementation of random forest with m=20:  0.04097253489419184
```

```
#m=20
#Splitting data to pass it to sklearn function
X_train1 = train_d.iloc[:, :-1].values
Y_train1= train_d.iloc[:, -1].values.reshape(-1,1)
X_test1 = test_d.iloc[:, :-1].values
Y_test1= test_d.iloc[:, -1].values.reshape(-1,1)

#Random forest using sklearn library
from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier(n_estimators=5,criterion='entropy',max_depth=20, min_samples_split=7,max_features=20) #bootstrap defa
model1.fit(X_train1,Y_train1.ravel())
skscore=model1.score(X_test1,Y_test1)
print("Accuracy using sklearn implementation of random forest with m=20: ",skscore)
```

```
Accuracy using sklearn implementation of random forest with m=20:  0.9333333333333333
```

Hence, the own implementation of random forest performs on par with sklearn built in random forest.

Qn.4b) Variation of sensitivity with respect to number of features to split.

Sensitivity is nothing but recall which can calculated from the confusion matrix.

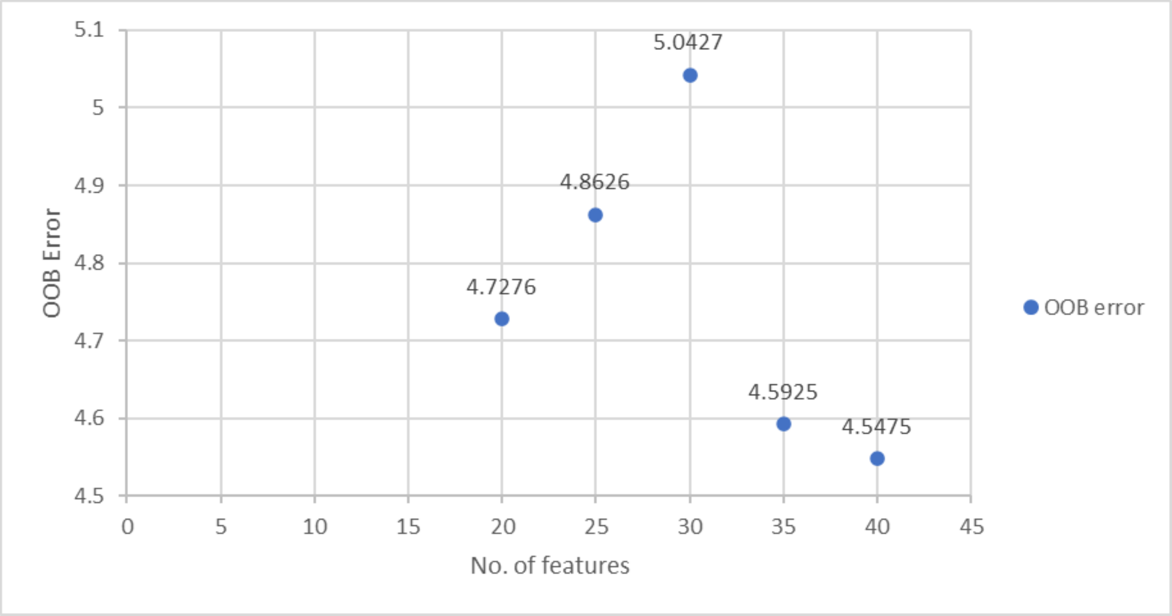
$$Sensitivity = \frac{TP}{TP + FN}$$

No. of features (m)	Sensitivity
20	0.9111498257839721
25	0.9041811846689896
30	0.8902439024390244
35	0.9111498257839721
40	0.9006968641114983

Qn.4c) OOB error vs No. of features(m)

No. of features (m)	OOB error (in %)
20	4.7276
25	4.8626
30	5.0427
35	4.5925
40	4.5475

The graph is plotted below:



Report by,

Prasanna Kumar R

sm21mtech14001