MACHINE LEARNING- CLASSIFICATION-ASSIGNMENTS

1.IDENTIFY THE PROBLEM STATEMENT

Stage 1: Machine learning

Stage 2: Supervised learning

Stage 3: classification(because dataset contains categorical data)

2.dataset contains 399 rows*25columns

3.PRE-PROCESSING

Categorical data into nominal data (using one hot encoding)

4.using algorithms to find the best model

- Using logistic regression

	precision	recall	f1-score	support
0	0.93	1.00	0.96	51
1	1.00	0.95	0.97	82
accuracy			0.97	133
macro avg	0.96	0.98	0.97	133
weighted avg	0.97	0.97	0.97	133

```
[22]: from sklearn.metrics import roc_auc_score

      roc_auc_score(y_test,grid.predict_proba(X_test)[:,1])
```

```
[22]: 0.9995217599234816
```

In svm classifier

```
from sklearn.metrics import f1_score
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
```

```
The f1_macro value for best parameter {'C': 10, 'gamma': 'auto', 'kernel': 'poly'}: 0.955283779067923
```

```
print(cm)
```

```
[[51  0]
 [ 6 76]]
```

```
print(clf_report)
```

```
              precision    recall  f1-score   support

           0       0.89      1.00      0.94        51
           1       1.00      0.93      0.96        82

    accuracy                           0.95       133
   macro avg       0.95      0.96      0.95       133
weighted avg       0.96      0.95      0.96       133
```

In decision tree classifier

```
from sklearn.metrics import f1_score
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
```

```
The f1_macro value for best parameter {'criterion': 'gini', 'max_features': 'sqrt', 'splitter': 'random'}: 0.9626932787797391
```

```
print(cm)
```

```
[[51  0]
 [ 5 77]]
```

```
print(clf_report)
```

```
              precision    recall  f1-score   support

           0       0.91      1.00      0.95        51
           1       1.00      0.94      0.97        82

    accuracy                           0.96       133
   macro avg       0.96      0.97      0.96       133
weighted avg       0.97      0.96      0.96       133
```

```
from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(X_test)[:,1])
```

```
0.9695121951219512
```

In random forest classifier

```
]: from sklearn.metrics import f1_score
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
```

```
The f1_macro value for best parameter {'criterion': 'gini', 'max_features': 'sqrt', 'n_estimators': 100}: 0.9924946382275899
```

```
]: print(cm)
```

```
[[51  0]
 [ 1 81]]
```

```
]: print(clf_report)
```

```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        51
           1       1.00      0.99      0.99        82

    accuracy                           0.99       133
   macro avg       0.99      0.99      0.99       133
weighted avg       0.99      0.99      0.99       133
```

```
]: from sklearn.metrics import roc_auc_score
roc_auc_score(y_test,grid.predict_proba(X_test)[:,1])
```

```
]: 0.9999999999999999
```

After checked with all algorithms, in random forest classifier we are getting highest f1_score as 0.99.

So we chose the best model using randomforest classifier