

untitled0

July 13, 2023

```
[70]: import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
```

```
[71]: df=pd.read_csv('/content/drive/MyDrive/Copy of finalplacementdata3.csv')
```

```
[72]: df
```

```
[72]:
```

	RegNo.	Quants	LogicalReasoning	Verbal	Programming	CGPA	\
0	T150054001	11	11	10	11	10.00	
1	T150054002	8	10	11	18	8.80	
2	T150054003	11	11	10	8	9.63	
3	T150054004	14	13	8	8	6.55	
4	T150054005	10	7	7	10	7.27	
..	
325	T150054326	22	25	24	23	9.35	
326	T150054327	25	23	21	22	8.51	
327	T150054328	21	23	24	25	8.82	
328	T150054329	22	25	24	21	8.59	
329	T150054330	21	21	23	22	8.50	

	Networking	CloudComp	WebServices	DataAnalytics	QualityAssurance	AI	\
0	4.3	8.2	8.4	8.3	8.1	8.0	
1	8.9	8.0	8.0	8.0	8.3	9.4	
2	8.4	8.0	8.0	8.0	8.0	5.4	
3	4.5	6.5	6.9	3.3	4.1	4.6	
4	3.1	5.6	6.7	4.6	2.7	4.4	
..	
325	8.4	9.4	8.8	8.9	9.1	8.4	
326	8.7	8.8	9.4	8.0	7.6	8.0	
327	7.8	9.3	9.0	8.6	8.1	8.7	
328	7.9	9.4	9.1	9.0	8.3	8.8	
329	7.9	9.2	9.3	8.9	7.7	8.1	

Placed

0	1
1	1
2	1
3	0
4	0
..	...
325	0
326	0
327	0
328	1
329	1

[330 rows x 13 columns]

```
[73]: df=df.drop('RegNo.',axis=1)
```

```
[78]: X=df.drop('Placed',axis=1)
      y=df['Placed']
```

```
[79]: from sklearn.model_selection import train_test_split
      from sklearn.metrics import accuracy_score,f1_score,precision_score
      X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=4)
```

```
[89]: models={
      'LogisticRegression':LogisticRegression(),
      'RandomForestClassifier':RandomForestClassifier(),
      'DecisionTreeClassifier':DecisionTreeClassifier()
    }
    for i in range(len(models)):
        model=list(models.values())[i]
        model.fit(X_train,y_train)

        # make predictions
        y_train_pre=model.predict(X_train)
        y_test_pre=model.predict(X_test)

        model_train_accuracy=accuracy_score(y_train,y_train_pre)
        model_train_f1=f1_score(y_train,y_train_pre)
        model_train_pre=precision_score(y_train,y_train_pre)

        # testing perfomance checking
        model_test_accuracy=accuracy_score(y_test,y_test_pre)
        model_test_f1=f1_score(y_test,y_test_pre)
        model_test_pre=precision_score(y_test,y_test_pre)

        print('\n')
        print(model)
```

```

print("="*15)
print("Performance of model while training")
print('-', "Accuracy = ", model_train_accuracy)
print('-', "F1_score = ", model_train_f1)
print('-', "Precision = ", model_train_pre)

print('\n'
)
print("Performance of model while testing")
print('-', "Accuracy = ", model_test_accuracy)
print('-', "F1_score = ", model_test_f1)
print('-', "Precision = ", model_test_pre)

print("_"*40)

```

/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
```

```
LogisticRegression()
```

```
=====
```

```
Performance of model while training
```

```
- Accuracy = 0.7765151515151515
- F1_score = 0.694300518134715
- Precision = 0.7528089887640449
```

```
Performance of model while testing
```

```
- Accuracy = 0.8333333333333334
- F1_score = 0.7027027027027026
- Precision = 0.6190476190476191
```

```
-----
```

```
RandomForestClassifier()
```

```
=====
```

```
Performance of model while training
```

```
- Accuracy = 1.0
- F1_score = 1.0
```

```
- Precision = 1.0
```

Perfomance of model while testing

```
- Accuracy = 0.803030303030303
- F1_score = 0.6666666666666667
- Precision = 0.5652173913043478
```

```
DecisionTreeClassifier()
```

=====

Perfomance of model while training

```
- Accuracy = 1.0
- F1_score = 1.0
- Precision = 1.0
```

Perfomance of model while testing

```
- Accuracy = 0.7121212121212122
- F1_score = 0.5777777777777777
- Precision = 0.4482758620689655
```

```
[90]: # Hyperparameter Training
rf_params={
    'max_depth': [5,8,10],
    'max_features': [5,7,"auto",8],
    'min_samples_split': [2,8,15,20],
    'n_estimators': [100,200,500,1000]
}
```

```
[91]: rf_params
```

```
[91]: {'max_depth': [5, 8, None, 10],
      'max_features': [5, 7, 'auto', 8],
      'min_samples_split': [2, 8, 15, 20],
      'n_estimators': [100, 200, 500, 1000]}
```

```
[93]: # again training the model
hy_rfmodel=[
    ('RF',RandomForestClassifier(),rf_params)
    ('')
]
```

```
[94]: hy_rfmodel
```

```
[94]: [('RF',
        RandomForestClassifier(),
        {'max_depth': [5, 8, None, 10],
         'max_features': [5, 7, 'auto', 8],
         'min_samples_split': [2, 8, 15, 20],
         'n_estimators': [100, 200, 500, 1000]})]
```

```
[ ]:
```