

### أسماء جروب المشروع :-

1- حسام إبراهيم فؤاد الحملي

2- صالح مؤنس علوان

3- شريف علاء جاد

4- أحمد سعد إبراهيم البدري

5- بيتر عادل فارس

---

### Random-Forest-Classifer Project

A very simple Random Forest Classifier implemented in python. The sklearn.ensemble library was used to import the RandomForestClassifier class. The object of the class was created. The following arguments was passed initially to the object:

```
n_estimators = 10
```

```
criterion = 'entropy'
```

The initial model was only given 10 decision tree, which resulted in a total of 10 incorrect prediction. Once the model was fitted with more the decision trees the number of incorrect prediction grew less.

It was found that a the optimal number of decision trees for this models to predict the answers was 200 decision trees. Hence the n\_estimator argument was given a final value of 200.

**Anything more that 200 will result in over-fitting and will lead further incorrect prediction.**

```
--- training data from kaggle
```

```
--- we use from ensemble Random forest classifier
```

## Code :

```
Import pandas as pd
```

```
df = pd.read_csv('Social_Network_Ads.csv') # data from kaggle
```

```
df
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le_Gender = LabelEncoder()
```

```
df['Gender'] = le_Gender.fit_transform(df['Gender']) ## 1 --> male ,,, 2 --> female
```

```
df
```

```
df['Purchased'] = df.Purchased # target
```

```
df
```

```
x = df.drop('Purchased',axis='columns')
```

```
y = df.Purchased
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2 , random_state=10)
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
model = RandomForestClassifier(n_estimators=200)
```

```
model.fit(x_train, y_train)
```

```
model.score(x_train, y_train)
```

```
model.score(x_test, y_test)
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.metrics import accuracy_score
```

```
scores = []
```

```
for k in range(1, 200):
```

```
    rfc = RandomForestClassifier(n_estimators=k)
```

```
    rfc.fit(x_train, y_train)
```

```
    y_pred = rfc.predict(x_test)
```

```
    scores.append(accuracy_score(y_test, y_pred))
```

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

```
plt.figure(figsize=(10,5) , dpi=150)
```

```
# plot the relationship between K and testing accuracy
```

```
# plt.plot(x_axis, y_axis)
```

```
plt.plot(range(1, 200), scores)
```

```
plt.xlabel('Value of n_estimators for Random Forest Classifier')
```

```
plt.ylabel('Testing Accuracy')
```

---

```
In [1]: import pandas as pd
```

executed in 1.10s, finished 21:32:21 2022-01-18

```
In [2]: df = pd.read_csv('Social_Network_Ads.csv')
df
```

executed in 85ms, finished 21:32:22 2022-01-18

```
Out[2]:
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15688575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

```
In [3]: from sklearn.preprocessing import LabelEncoder
le_Gender = LabelEncoder()

df['Gender'] = le_Gender.fit_transform(df['Gender']) ## 1 --> male , , 2 --> female
```

executed in 2.04s, finished 21:32:29 2022-01-18

```
In [4]: df
```

executed in 59ms, finished 21:32:29 2022-01-18

```
Out[4]:
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	1	19	19000	0
1	15810944	1	35	20000	0
2	15688575	0	26	43000	0
3	15603246	0	27	57000	0
4	15804002	1	19	76000	0
...	...	...	...	...	...
395	15691863	0	46	41000	1
396	15706071	1	51	23000	1
397	15654296	0	50	20000	1
398	15755018	1	36	33000	0
399	15594041	0	49	36000	1

400 rows × 5 columns

```
In [9]: df
```

executed in 45ms, finished 20:49:34 2022-01-11

Out[9]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	1	19	19000	0
1	15810944	1	35	20000	0
2	15668575	0	26	43000	0
3	15603246	0	27	57000	0
4	15804002	1	19	76000	0
...	...	...	...	...	...
395	15691863	0	46	41000	1
396	15706071	1	51	23000	1
397	15654296	0	50	20000	1
398	15755018	1	36	33000	0
399	15594041	0	49	38000	1

400 rows x 5 columns

```
In [29]: df['Purchased'] = df.Purchased # target
df
```

executed in 42ms, finished 21:14:02 2022-01-18

Out[29]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	1	19	19000	0
1	15810944	1	35	20000	0
2	15668575	0	26	43000	0
3	15603246	0	27	57000	0
4	15804002	1	19	76000	0
...	...	...	...	...	...
395	15691863	0	46	41000	1
396	15706071	1	51	23000	1
397	15654296	0	50	20000	1

```
In [5]: x = df.drop('Purchased',axis='columns')
        y = df.Purchased
        executed in 12ms, finished 21:32:46 2022-01-18
```

```
In [6]: from sklearn.model_selection import train_test_split
        x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2 , random_state=10)
        executed in 98ms, finished 21:32:48 2022-01-18
```

```
In [24]: from sklearn.ensemble import RandomForestClassifier
         model = RandomForestClassifier(n_estimators=200)
         model.fit(x_train, y_train)
         executed in 458ms, finished 21:34:26 2022-01-18
```

```
Out[24]: RandomForestClassifier(n_estimators=200)
```

```
In [25]: model.score(x_train, y_train)
         executed in 90ms, finished 21:34:27 2022-01-18
```

```
Out[25]: 1.0
```

```
In [26]: model.score(x_test, y_test)
         executed in 83ms, finished 21:34:43 2022-01-18
```

```
Out[26]: 0.9375
```

```
In [28]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import accuracy_score
         scores = []
         for k in range(1, 200):
             rfc = RandomForestClassifier(n_estimators=k)
             rfc.fit(x_train, y_train)
             y_pred = rfc.predict(x_test)
             scores.append(accuracy_score(y_test, y_pred))

         import matplotlib.pyplot as plt
         %matplotlib inline
         plt.figure(figsize=(10,5) , dpi=150)
         # plot the relationship between K and testing accuracy
         # plt.plot(x_axis, y_axis)
         plt.plot(range(1, 200), scores)
         plt.xlabel('Value of n_estimators for Random Forest Classifier')
         plt.ylabel('Testing Accuracy')
         executed in 51.8s, finished 21:44:39 2022-01-18
```

```
Out[28]: Text(0, 0.5, 'Testing Accuracy')
```

