import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

dataset=pd.read\_csv("Social\_Network\_Ads.csv")

dataset=pd.get\_dummies(dataset,drop\_first=True)

dataset

| **User ID** | **Age** | **EstimatedSalary** | **Purchased** | **Gender\_Male** |
| --- | --- | --- | --- | --- |
| 0 | 15624510 | 19 | 19000 | 0 | 1 |
| 1 | 15810944 | 35 | 20000 | 0 | 1 |
| 2 | 15668575 | 26 | 43000 | 0 | 0 |
| 3 | 15603246 | 27 | 57000 | 0 | 0 |
| 4 | 15804002 | 19 | 76000 | 0 | 1 |
| ... | ... | ... | ... | ... | ... |
| 395 | 15691863 | 46 | 41000 | 1 | 0 |
| 396 | 15706071 | 51 | 23000 | 1 | 1 |
| 397 | 15654296 | 50 | 20000 | 1 | 0 |
| 398 | 15755018 | 36 | 33000 | 0 | 1 |
| 399 | 15594041 | 49 | 36000 | 1 | 0 |

400 rows × 5 columns

In [5]:

dataset.columns

Out[5]:

Index(['User ID', 'Age', 'EstimatedSalary', 'Purchased', 'Gender\_Male'], dtype='object')

In [12]:

independent**=**dataset[['User ID', 'Age', 'EstimatedSalary', 'Gender\_Male']]

dependent**=**dataset[['Purchased']]

In [13]:

independent.shape

Out[13]:

(400, 4)

In [14]:

independent

Out[14]:

|  | **User ID** | **Age** | **EstimatedSalary** | **Gender\_Male** |
| --- | --- | --- | --- | --- |
| 0 | 15624510 | 19 | 19000 | 1 |
| 1 | 15810944 | 35 | 20000 | 1 |
| 2 | 15668575 | 26 | 43000 | 0 |
| 3 | 15603246 | 27 | 57000 | 0 |
| 4 | 15804002 | 19 | 76000 | 1 |
| ... | ... | ... | ... | ... |
| 395 | 15691863 | 46 | 41000 | 0 |
| 396 | 15706071 | 51 | 23000 | 1 |
| 397 | 15654296 | 50 | 20000 | 0 |
| 398 | 15755018 | 36 | 33000 | 1 |
| 399 | 15594041 | 49 | 36000 | 0 |

400 rows × 4 columns

In [8]:

dataset['Purchased'].value\_counts()

Out[8]:

0 257

1 143

Name: Purchased, dtype: int64

In [9]:

dependent

Out[9]:

|  | **Purchased** |
| --- | --- |
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| ... | ... |
| 395 | 1 |
| 396 | 1 |
| 397 | 1 |
| 398 | 0 |
| 399 | 1 |

400 rows × 1 columns

In [15]:

*#split in to test set and training set*

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

x\_train,x\_test,y\_train,y\_test**=**train\_test\_split(independent,dependent,test\_size**=**1**/**3,random\_state**=**0)

*#creating the model*

**from** sklearn.ensemble **import** RandomForestClassifier

classifier **=** RandomForestClassifier(n\_estimators **=** 10, criterion **=** 'entropy', random\_state **=** 0)

classifier.fit(x\_train, y\_train)

C:\Users\admin\Anaconda3\lib\site-packages\ipykernel\_launcher.py:4: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples,), for example using ravel().

after removing the cwd from sys.path.

Out[16]:

RandomForestClassifier(criterion='entropy', n\_estimators=10, random\_state=0)

In [17]:

y\_pred**=**classifier.predict(x\_test)

In [19]:

*# from sklearn.metrics import confusion\_matrix*

*# cm= confusion\_matrix(y\_test, y\_pred)*

​

**from** sklearn.metrics **import** confusion\_matrix

cm **=** confusion\_matrix(y\_test, y\_pred)

In [22]:

print(cm)

[[79 6]

[ 6 43]]

In [29]:

*# from sklearn.metrics import classification\_report*

*# clf\_report = classification\_report(y\_test, y\_pred)*

​

**from** sklearn.metrics **import** classification\_report

clf\_report**=**classification\_report(y\_test,y\_pred)

In [30]:

print(clf\_report)

precision recall f1-score support

0 0.93 0.93 0.93 85

1 0.88 0.88 0.88 49

accuracy 0.91 134

macro avg 0.90 0.90 0.90 134

weighted avg 0.91 0.91 0.91 134

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| 0.00 | 0.93 | 0.93 | 0.93 | 85 |
| 1.00 | 0.88 | 0.88 | 0.88 | 49 |
|  |  |  |  |  |
| Accuracy |  |  | 0.91 | 134 |
| macro avg | 0.90 | 0.90 | 0.90 | 134 |
| weighted avg | 0.91 | 0.91 | 0.91 | 134 |

++++++++++++++++++++++++++++++++++++++++++++++++++

|  | | **Actual** | |
| --- | --- | --- | --- |
| **Dog** | **Not Dog** |
| **Predicted** | **Dog** | True Positive (TP =5) | False Positive (FP=1) |
| **Not Dog** | False Negative (FN =1) | True Negative (TN=3) |

[[79 6]

[ 6 43]]

TP=79, TN=43, FP=6, FN=6

1. What is the Accuracy of the model

🡪Accuracy is used to measure the performance of the model. It is the ratio of Total correct instances to the total instances.

Accuracy=(TP+TN)/(FP+FNTP+TN​)

=(79+43)/(79+43+6+6=0.910448

1. What is the Precision of the model.

🡪Precision is a measure of how accurate a model’s positive predictions are. It is defined as the ratio of true positive predictions to the total number of positive predictions made by the model.

Precision= TP/TP+FP

=79/(6+43)= 1.612244898

1. What I the Recall value of the model?

🡪Recall measures the effectiveness of a classification model in identifying all relevant instances from a dataset. It is the ratio of the number of true positive (TP) instances to the sum of true positive and false negative (FN) instances.

Recall= TP/TP+FN

=79/(79+6)= 0.929411765

1. What is the F1-Score of the model

🡪F1-score is used to evaluate the overall performance of a classification model. It is the harmonic mean of precision and recall,

F1-Score=(2\*Precision\*recall)/(Precision+Recall)

=(2\*1.612244898\*0.929411765)/(1.612244898+0.929411765)

=1.179104478

1. What is the specificity of the model?

🡪Specificity is another important metric in the evaluation of classification models, particularly in binary classification. It measures the ability of a model to correctly identify negative instances. Specificity is also known as the True Negative Rate.

Specificity =TN/(TN+FP)

=43/(43+6)= 0.8755

1. Type 1 and Type 2 error

* Type 1 error occurs when the model predicts a positive instance, but it is actually negative. Precision is affected by false positives, as it is the ratio of true positives to the sum of true positives and false positives.

=FP/TN+FP

=6/(43+6)=0.1224489

* Type 2 error occurs when the model fails to predict a positive instance. Recall is directly affected by false negatives, as it is the ratio of true positives to the sum of true positives and false negatives.

=FP/TP+FN

=6/(79+6)=0.070588

|  |  |
| --- | --- |
| Metrics for Confusion Matrix | RF\_CLASS values |
| Accuracy | 0.910448 |
| Precision | 1.61224 |
| Recall | 0.929411765 |
| F1\_Score | 1.179104478 |
| Specificity | 0.8755 |
| Sensitivity | 0.8755 |
| Type1 error | 0.1224489 |
| Type 2 error | 0.070588 |

Questions

1. What is the overall performance of the model

* Overall performance refers to accuracy and value for Random forest model is 0.910448

1. What is the correct classification of not purchase in RF model?

* 0.93

1. What is the correct classification of purchase in RF model?

* 0.88

1. What is the overall performance of not purchase?

* 0.93

1. What is the overall performance of purchase?

* 0.88

1. What is the F1 score of not purchasd?

* 0.93

1. What is the F1 score of purchased?

* 0.88