*#importing the libraries*

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

In [2]:

dataset**=**pd.read\_csv("C:\Prakash\AI\Learning\Machine\_learning\Classification\Random\_forest\_classifier\Social\_Network\_Ads.csv")

In [3]:

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

In [4]:

dataset

Out[4]:

|  | **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 [6]:

dataset**=**dataset.drop("User ID",axis**=**1)

In [ ]:

dataset

In [7]:

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

Out[7]:

0 257

1 143

Name: Purchased, dtype: int64

In [14]:

dataset.columns

Out[14]:

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

In [15]:

independ**=**dataset[["Age", "EstimatedSalary", "Gender\_Male"]]

depend**=**dataset[["Purchased"]]

In [16]:

independ.shape

Out[16]:

(400, 3)

In [17]:

depend

Out[17]:

|  | **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 [20]:

*#split in to training and test set*

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

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

In [24]:

from sklearn.neighbors import KNeighborsClassifier

classifier=KNeighborsClassifier(n\_neighbors=7,metric='minkowski',p=2)

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

KNeighborsClassifier(n\_neighbors=7)

In [15]:

In [26]:

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

from sklearn.metrics import confusion\_matrix

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

print(cm)

[[79 6]

[16 33]]

from sklearn.metrics import classification\_report

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

n [36]:

print(clf\_report)

precision recall f1-score support

0 0.83 0.93 0.88 85

1 0.85 0.67 0.75 49

accuracy 0.84 134

macro avg 0.84 0.80 0.81 134

weighted avg 0.84 0.84 0.83 134

In [39]::

classifier.predict([[46,41000,0]])

C:\Users\admin\Anaconda3\lib\site-packages\sklearn\base.py:442: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names

"X does not have valid feature names, but"

Out[45]:

array([0], dtype=int64)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-score | Support |
| 0 | 0.83 | 0.93 | 0.88 | 85 |
| 1 | 0.85 | 0.67 | 0.75 | 49 |
|  |  |  |  |  |
| Accuracy |  |  | 0.84 | 134 |
| Macro avg | 0.84 | 0.80 | 0.81 | 134 |
| Weighted avg | 0.84 | 0.84 | 0.83 | 134 |

|  | | **Actual** | |
| --- | --- | --- | --- |
| **Purchased** | **Not Purchased** |
| **Predicted** | **Purchased** | True Positive (TP =79) | False Positive (FP=6) |
| **Not Purchased** | False Negative (FN =16) | True Negative (TN=33) |

TP=79, TN=33, FP=6, FN=16

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+FN+TP+TN​)=(79+33)/(79+33+6+16)= 0.835820896

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/(79+6)= 0.929411765

1. What I the Recall/Sensitivity 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+16)= 0.831578947

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\* 0.929411765\*0.831578947)/( 0.929411765+ 0.831578947)= 0.877777778

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)=33/(33+6)= 0.846153846

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/(33+6)= 0.153846154

* 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+16)= 0.063157895

|  |  |
| --- | --- |
| Metrics for Confusion Matrix | RF\_CLASS values |
| Accuracy | 0.835821 |
| Precision | 0.929412 |
| Recall | 0.831579 |
| F1\_Score | 0.877778 |
| Specificity | 0.846154 |
| Type1 error | 0.153846 |
| Type 2 error | 0.063158 |
| Macro Average | 0.879589 |

Questions

1. What is the overall performance of the model

* Overall performance refers to accuracy and value for Random forest model is 0.87

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

* 0.89

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

* 0.84

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