Name : **Srinivasan JP** Reg No: **21MIS1044**

K-Nearest Neighbors

- Data set choosen is Iphone purchase records, based on the gender, age, salary predicting the person will buy the Iphone or not

In [ ]:
```
#imports
import seaborn as sns
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
```

In [ ]:
```
#reading dataset
df = pd.read_csv("./iphone_purchase_records.csv")
df.head()
```

Out[ ]:

	Gender	Age	Salary	Purchase Iphone
0	Male	19	19000	0
1	Male	35	20000	0
2	Female	26	43000	0
3	Female	27	57000	0
4	Male	19	76000	0

In [ ]:
```
#convert gender to binary nominal variable
df['Gender'].replace({'Male':1, 'Female':0},inplace=True)
df.head()
```

Out[ ]:

	Gender	Age	Salary	Purchase Iphone
0	1	19	19000	0
1	1	35	20000	0
2	0	26	43000	0
3	0	27	57000	0
4	1	19	76000	0

Spliting the independent and dependent variables from the dataFrame

In [ ]:
```
x = df.iloc[:,:3].values
y = df.iloc[:,-1].values
```

Spliting the dataset into train and test set using sklearn train_test_split train set size is 70% test set size is 30%

In [ ]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,train_
```

In [ ]:
```
#feature scaling
ss = StandardScaler()
x_train = ss.fit_transform(x_train)
x_test = ss.transform(x_test)
```

Fitting the K-NN classifier to the training set

In [ ]:
```
classifier = KNeighborsClassifier(n_neighbors=5, metric='mink
classifier.fit(x_train, y_train)
```

Out[ ]:

KNeighborsClassifier

```
KNeighborsClassifier()
```

Predicting the test data using the trained classifier

In [ ]:
```
y_predict = classifier.predict(x_test)
print(y_predict)
```

```
[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0
0 1 0 0 0 0
 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 0
0 0 1 0 0 1
 0 0 0 0 1 1 1 1 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 1 0 1 1
1 0 0 0 0 0
 0 0 1 1 1 1 0 0 1]
```

Building the confusion matrix to get the accuracy of the classifier

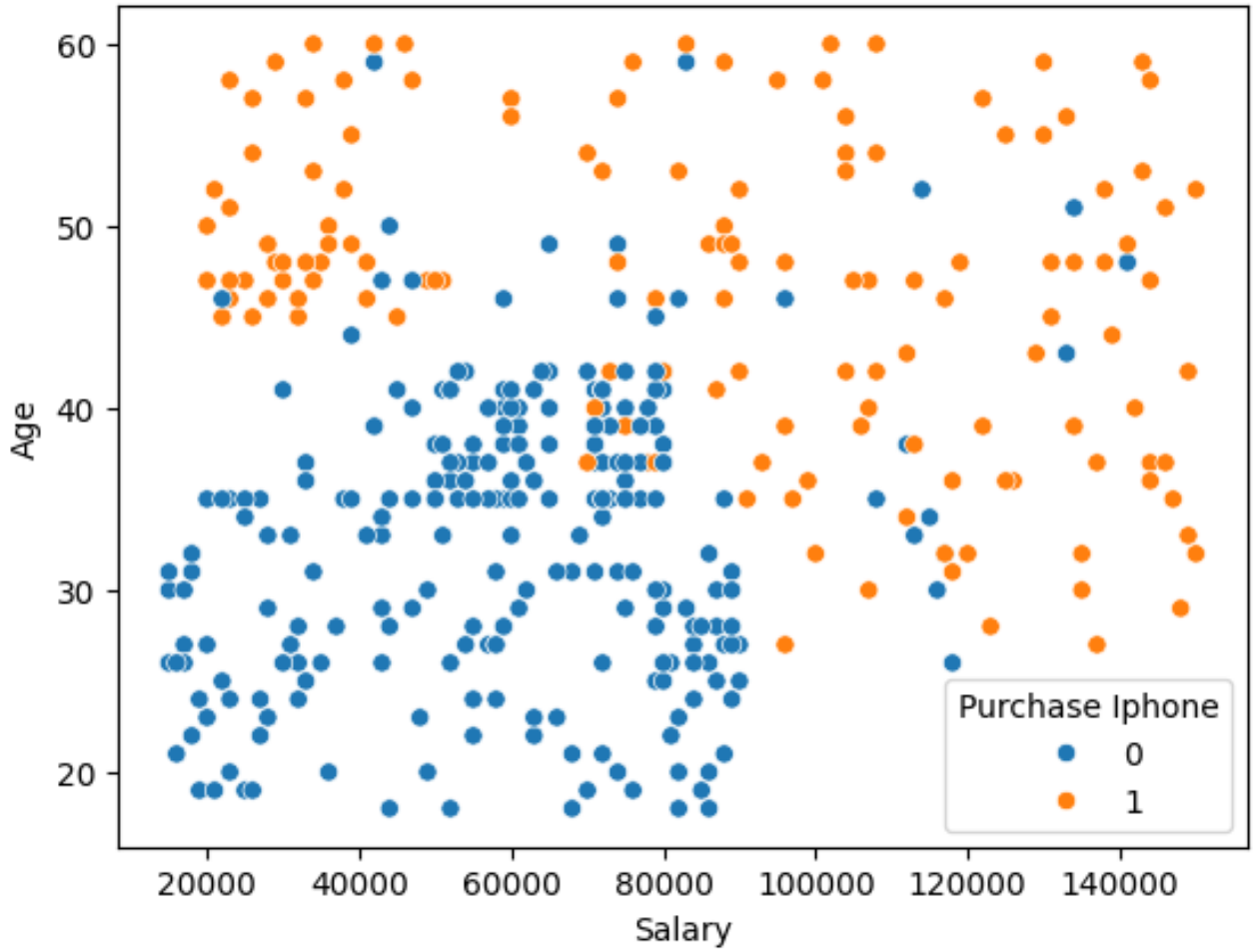
In [ ]:
```
accuracy = accuracy_score(y_test, y_predict)
print(f"Accuracy: {accuracy*100}")
matrix = confusion_matrix(y_test, y_predict)
correct_prediction = matrix[0,0] + matrix[1,1]
wrong_prediction = matrix[0,1] + matrix[1,0]
print(f"Corrrect Prediction {correct_prediction}", f"Wrong Pre
```

```
Accuracy: 91.66666666666666
Corrrect Prediction 110
Wrong Prediction 10
```

Visualizing the training set result using matplotlib

In [ ]:
```
sns.scatterplot(x=df["Salary"], y=df["Age"], hue=df["Purchase
```

Out[ ]:
```
<Axes: xlabel='Salary', ylabel='Age'>
```


60
50
40
30
20
Age
20000
40000
60000
80000
100000
120000
140000
Salary
Purchase Iphone
0
1