In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
#read the data
df=pd.read_csv("Social_Network_Ads.csv")
```

In [3]:
```
df
```

Out[3]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	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 [4]:
```
df.drop(columns="User ID",inplace=True)
```

In [5]:
```
df
```

Out[5]:

	Gender	Age	EstimatedSalary	Purchased
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
...	...	...	...	...
395	Female	46	41000	1
396	Male	51	23000	1
397	Female	50	20000	1
398	Male	36	33000	0
399	Female	49	36000	1

400 rows × 4 columns

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Gender           400 non-null    object
 1   Age              400 non-null    int64 
 2   EstimatedSalary  400 non-null    int64 
 3   Purchased        400 non-null    int64 
dtypes: int64(3), object(1)
memory usage: 12.6+ KB
```

In [7]:
```
df.describe()
```

Out[7]:

	Age	EstimatedSalary	Purchased
count	400.000000	400.000000	400.000000
mean	37.655000	69742.500000	0.357500
std	10.482877	34096.960282	0.479864
min	18.000000	15000.000000	0.000000
25%	29.750000	43000.000000	0.000000
50%	37.000000	70000.000000	0.000000
75%	46.000000	88000.000000	1.000000
max	60.000000	150000.000000	1.000000

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

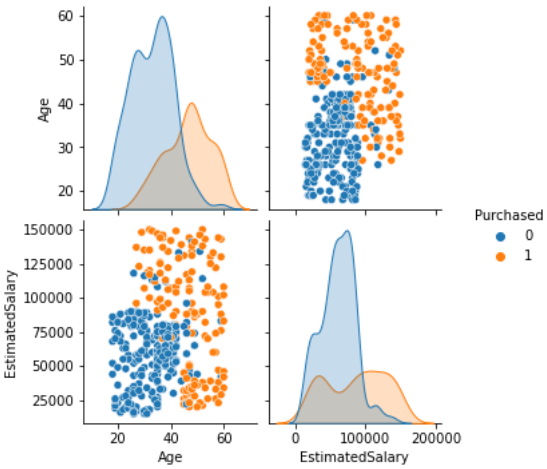
In [10]:
```
df["Purchased"].value_counts()
```

Out[10]:
```
0    257
1    143
Name: Purchased, dtype: int64
```

In [11]:
```
#THIS CONTAINS TWO CLASSES HENCE IT IS THE BINARY CLSSIFICATION ALGORITHM PROPLEM
#THE DATA IS BALANCED
```

In [13]:
```
#showing various plots of distribusion of data
sns.pairplot(df,hue="Purchased")
```

Out[13]:
```
<seaborn.axisgrid.PairGrid at 0x15e2b790910>
```



In [15]:
```
df.head()
```

Out[15]:

	Gender	Age	EstimatedSalary	Purchased
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 [14]:
```
#seperate x and y
```

In [16]:
```
x=df.iloc[:,1:-1]
```

In [17]:
```
x
```

Out[17]:

	Age	EstimatedSalary
0	19	19000
1	35	20000
2	26	43000
3	27	57000
4	19	76000
...	...	...
395	46	41000
396	51	23000
397	50	20000
398	36	33000
399	49	36000

400 rows × 2 columns

In [18]:
```
y=df["Purchased"]
```

In [19]:
```
y
```

Out[19]:
```
0      0
1      0
2      0
3      0
4      0
      ..
395    1
396    1
397    1
398    0
399    1
Name: Purchased, Length: 400, dtype: int64
```

In [20]:
```
#splitting the data for testing and training
```

In [22]:
```
from sklearn.model_selection import train_test_split
```

In [23]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [24]:
```
#import the model and packages
from sklearn.neighbors import KNeighborsClassifier
```

In [26]:
```
#stp2: create instance of model
knn=KNeighborsClassifier(n_neighbors=3)
```

In [27]:
```
#MODEL FITTING
```

In [28]:
```
knn.fit(x_train,y_train)
```

Out[28]:
```
KNeighborsClassifier(n_neighbors=3)
```

In [29]:
```
ypred=knn.predict(x_test)
```

In [30]:
```
#evaluation model
```

In [31]:
```
from sklearn.metrics import accuracy_score
```

In [32]:
```
accuracy_score(y_test,ypred)
```

Out[32]:
```
0.78
```

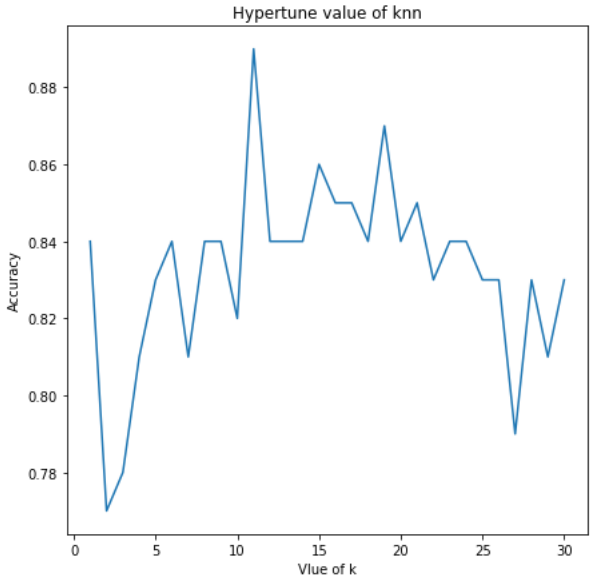
In [33]:
```
#try to find best value of k:Hypertune
```

In [40]:
```
ac_list=[]
for i in range(1,31):
    #stp2: create instance of model
    knn=KNeighborsClassifier(n_neighbors=i)
    #model fitting
    knn.fit(x_train,y_train)
    ypred=knn.predict(x_test)
    ac= accuracy_score(y_test,ypred)
    ac_list.append(ac)
```

In [41]:
```
ac_list
```

Out[41]:
```
[0.84,
 0.77,
 0.78,
 0.81,
 0.83,
 0.84,
 0.81,
 0.84,
 0.84,
 0.82,
 0.89,
 0.84,
 0.84,
 0.84,
 0.86,
 0.85,
 0.85,
 0.84,
 0.87,
 0.84,
 0.85,
 0.83,
 0.84,
 0.84,
 0.83,
 0.83,
 0.79,
 0.83,
 0.81,
 0.83]
```

In [43]:
```
plt.figure(figsize=(7,7))
plt.title("Hypertune value of knn")
plt.plot(range(1,31),ac_list)
plt.xlabel("Vlue of k")
plt.ylabel("Accuracy")
plt.show()
```



In [45]:
```
#import the model and packages
from sklearn.neighbors import KNeighborsClassifier

#stp2: create instance of model
knn=KNeighborsClassifier(n_neighbors=11)

#MODEL FITTING
knn.fit(x_train,y_train)

ypred=knn.predict(x_test)

#evaluation model
from sklearn.metrics import accuracy_score

accuracy_score(y_test,ypred)
```

Out[45]:
```
0.89
```

In [46]:
```
np.sqrt(len(y_test))
```

Out[46]:
```
10.0
```

In [ ]:
```

```