untitled1

March 28, 2025

[1]:
```
#Name: SHIVANI GADKARI
#Roll no: 13342
```

[48]:
```
import pandas as pd
import numpy as np
import matplotlib as plt
```

[49]:
```
df=pd.read_csv("social_network_ads.csv")
```

[50]:
```
df
```

[50]:

	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 x 5 columns]

[51]:
```
df.columns
```

[51]:
```
Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'],
dtype='object')
```

[52]:
```
df.isnull()
```

[52]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False

```
3      False   False  False            False      False
4      False   False  False            False      False
..       …       …    …                …          …
395    False   False  False            False      False
396    False   False  False            False      False
397    False   False  False            False      False
398    False   False  False            False      False
399    False   False  False            False      False

[400 rows x 5 columns]
```

[57]:
```
df
```

[57]:
```
      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 x 5 columns]
```

[58]:
```
from sklearn.model_selection import train_test_split
```

[59]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(x, y,test_size=0.
 ↪4,random_state=10)
```

[61]:
```
from sklearn.linear_model import LogisticRegression
 print(X_train.head())
```

```
      User ID  Gender  Age  EstimatedSalary
60   15814004       1   27            20000
21   15736760       0   47            49000
299  15747043       1   46           117000
106  15706185       0   26            35000
139  15741094       1   19            25000
```

[63]:
```
X_train= pd.get_dummies(X_train,drop_first=True)
```

[89]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
X = pd.get_dummies(df.drop(columns=['Purchased']), drop_first=True)
y = df['Purchased']
X_train,X_test, Y_train,Y_test= train_test_split(X,y,test_size=0.
 ↪2,random_state=42)
X_train= pd.get_dummies(X_train,drop_first=True)
X_test =pd.get_dummies(X_test,drop_first=True)
logreg =LogisticRegression()
logreg.fit(X_train,Y_train)
```

[89]:
```
LogisticRegression()
```

[65]:
```
Y_pred =logreg.predict(X_test)
 print("Predictions:", Y_pred)
```

```
Predictions: [0 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 0 1 0 1 1 0 1 0 0 0 1
0 1 0 0
 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0
 0 0 1 1 0 0]
```

[66]:
```
import sklearn
from sklearn.linear_model import LogisticRegression
logreg =LogisticRegression()
model=logreg.fit(X_train,Y_train)
```

[67]:
```
Ytrain_pred= logreg.predict(X_train)
 Ytest_pred= logreg.predict(X_test)
```

[68]:
```
df=pd.DataFrame(Ytrain_pred,Y_train)
 df=pd.DataFrame(Ytest_pred,Y_test)
```

[69]:
```
from sklearn.metrics import␣
 ↪precision_score,confusion_matrix,accuracy_score,recall_score
cm =confusion_matrix(Y_test,Y_pred)
cm =confusion_matrix(Y_test,Y_pred)
cm =confusion_matrix(Y_test,Y_pred)
print("ConfusionMatrix:\n",cm)
```

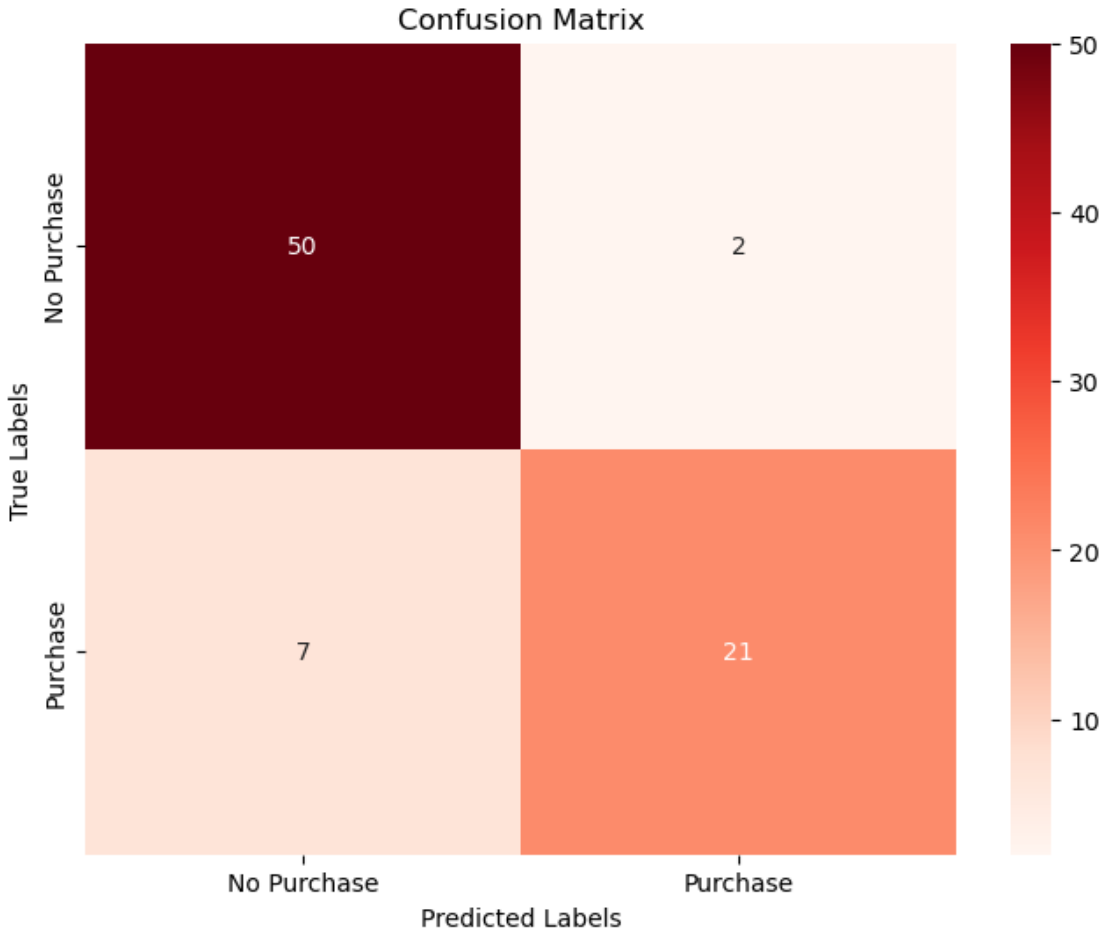
```
ConfusionMatrix:
 [[50  2]
 [ 7 21]]
```

[70]:
```
print("Accuracy:", accuracy_score(Y_test, Y_pred))
 print("Precision:", precision_score(Y_test, Y_pred, average='weighted'))
 print("Recall:", recall_score(Y_test, Y_pred, average='weighted'))
```

```
Accuracy: 0.8875
Precision: 0.8897406559877956
Recall: 0.8875
```

[71]:
```
import matplotlib.pyplot as plt
 import seaborn as sns
 from sklearn.metrics import confusion_matrix
```

[91]:
```
# Generate confusion matrix
 cm = confusion_matrix(Y_test, Y_pred)
 # Plot confusion matrix with correct labels
 labels = ['No Purchase', 'Purchase'] # Your class labels
 plt.figure(figsize=(8, 6))
 sns.heatmap(cm,␣
 ↪annot=True,fmt='d',cmap='Reds',xticklabels=labels,yticklabels=labels)
 plt.xlabel('Predicted Labels')
 plt.ylabel('True Labels')
 plt.title('Confusion Matrix')
 plt.show()
```



[80]:
```
print(df.head())
```

```
           0
Purchased
0          0
1          1
0          0
1          1
0          0
```

[81]:
```
print(df.columns)  # Make sure 'Gender' and 'Purchased' exist
```

```
RangeIndex(start=0, stop=1, step=1)
```

[82]:
```
df = pd.read_csv('social_network_ads.csv')  # Adjust the file path
```

[83]:
```
df.columns
```

[83]:
```
Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'],
dtype='object')
```

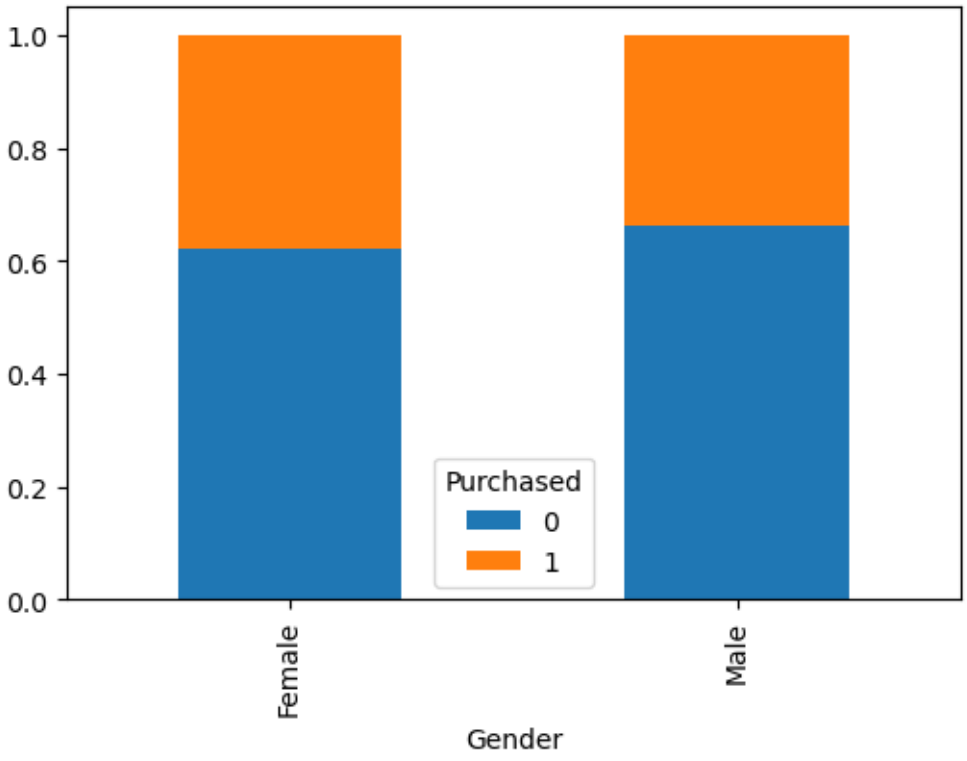
[84]:
```
df.head()
```

[84]:

	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

[90]:
```
ct=pd.crosstab(df['Gender'],df['Purchased'],normalize='index')
print(ct)
ct.plot.bar(figsize=(6,4),stacked=True)
plt.show()
```

```
Purchased         0         1
Gender
Female     0.622549  0.377451
Male       0.663265  0.336735
```


1.0
0.8
0.6
0.4
0.2
0.0
Purchased
0
1
Female
Male
Gender

[ ]: