Assignment No 5

```python
import pandas as pd
import numpy as np
import seaborn as sns
import io
import matplotlib.pyplot as plt

from google.colab import files
uploaded=files.upload()
```

```
<IPython.core.display.HTML object>

Saving Social_Network_Ads.csv to Social_Network_Ads.csv
```

```python
df=pd.read_csv(io.BytesIO(uploaded['Social_Network_Ads.csv']))

print(df.head(5))
```

```
    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
```

```python
sns.countplot('Gender', hue='Gender', data=df)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  FutureWarning
```
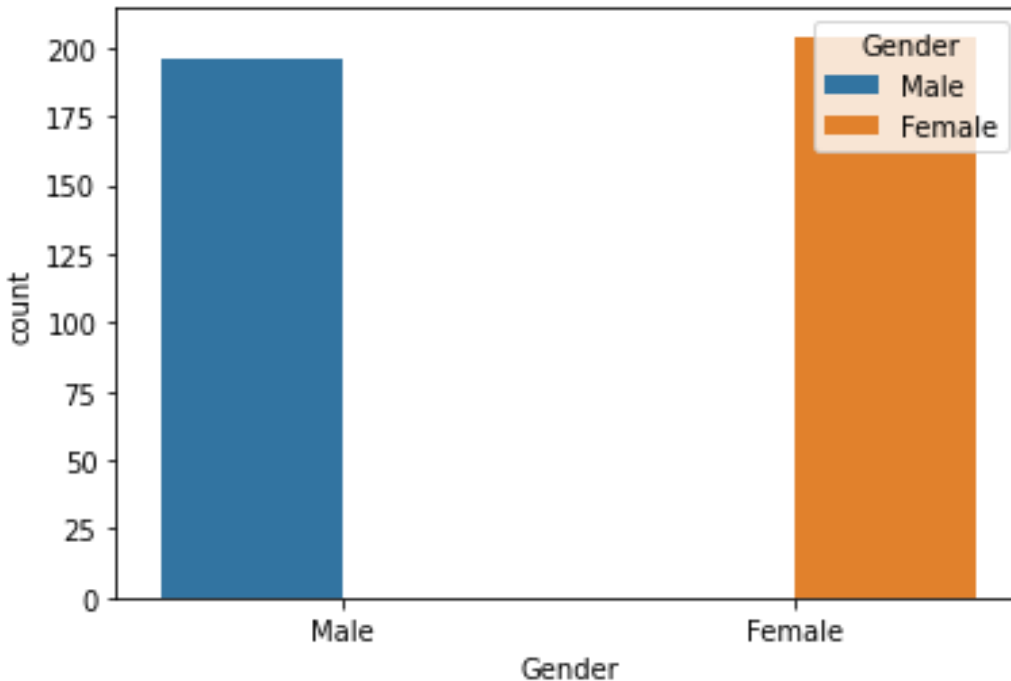
```
print(df.isnull().sum())
```

```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix,accuracy_score
```

```
x = df.drop('Gender', axis=1)
y = df['Gender']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33,
random_state=1)
logmodel = LogisticRegression()
logmodel.fit(x_train, y_train)
```

```
predictions = logmodel.predict(x_test)
print(classification_report(y_test, predictions))
print(confusion_matrix(y_test, predictions))
print(accuracy_score(y_test, predictions))
```

```
             precision    recall  f1-score   support

     Female       0.53      0.58      0.56        69
```

```
        Male        0.49        0.44        0.47          63

    accuracy                                0.52         132
   macro avg        0.51        0.51        0.51         132
weighted avg        0.51        0.52        0.51         132

[[40 29]
 [35 28]]
0.5151515151515151
```

```
sns.heatmap(pd.DataFrame(confusion_matrix(y_test,predictions)))
plt.show()
```