In [203…

```
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
```

In [204…

```
data = pd.read_csv("Transformed Data Set - Sheet1.csv")
```

In [205…

```
data.head()
```

Out[205]:

	Favorite Color	Favorite Music Genre	Favorite Beverage	Favorite Soft Drink	Gender
0	Cool	Rock	Vodka	7UP/Sprite	F
1	Neutral	Hip hop	Vodka	Coca Cola/Pepsi	F
2	Warm	Rock	Wine	Coca Cola/Pepsi	F
3	Warm	Folk/Traditional	Whiskey	Fanta	F
4	Cool	Rock	Vodka	Coca Cola/Pepsi	F

In [206…

```
data.shape
```

Out[206]:

```
(66, 5)
```

In [207…

```
data.isna().sum()
```

Out[207]:

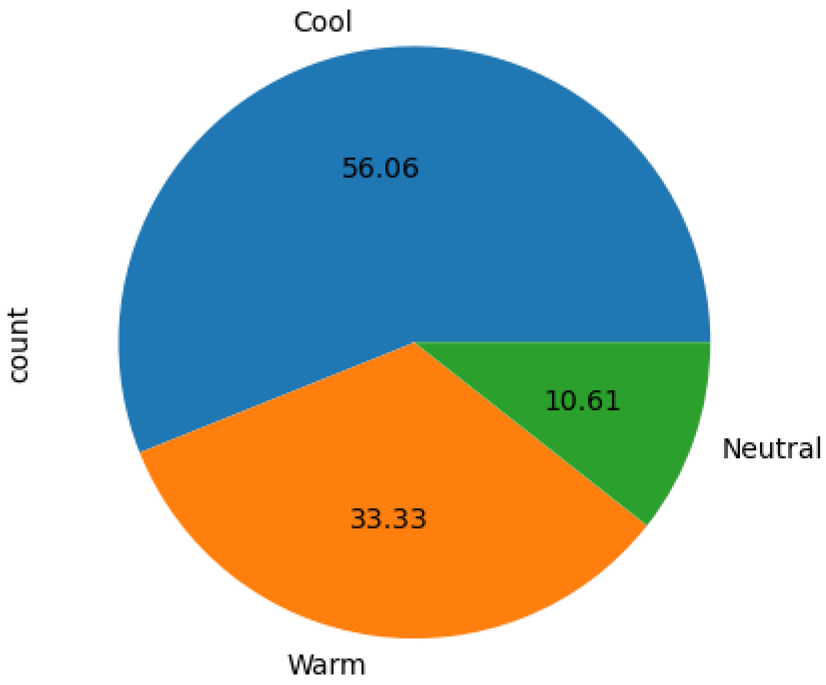
```
Favorite Color          0
Favorite Music Genre    0
Favorite Beverage       0
Favorite Soft Drink     0
Gender                  0
dtype: int64
```

In [208…

```
data["Favorite Color"].value_counts().plot(kind = "pie" , autopct = "%.2f")
```

Out[208]:

```
<Axes: ylabel='count'>
```

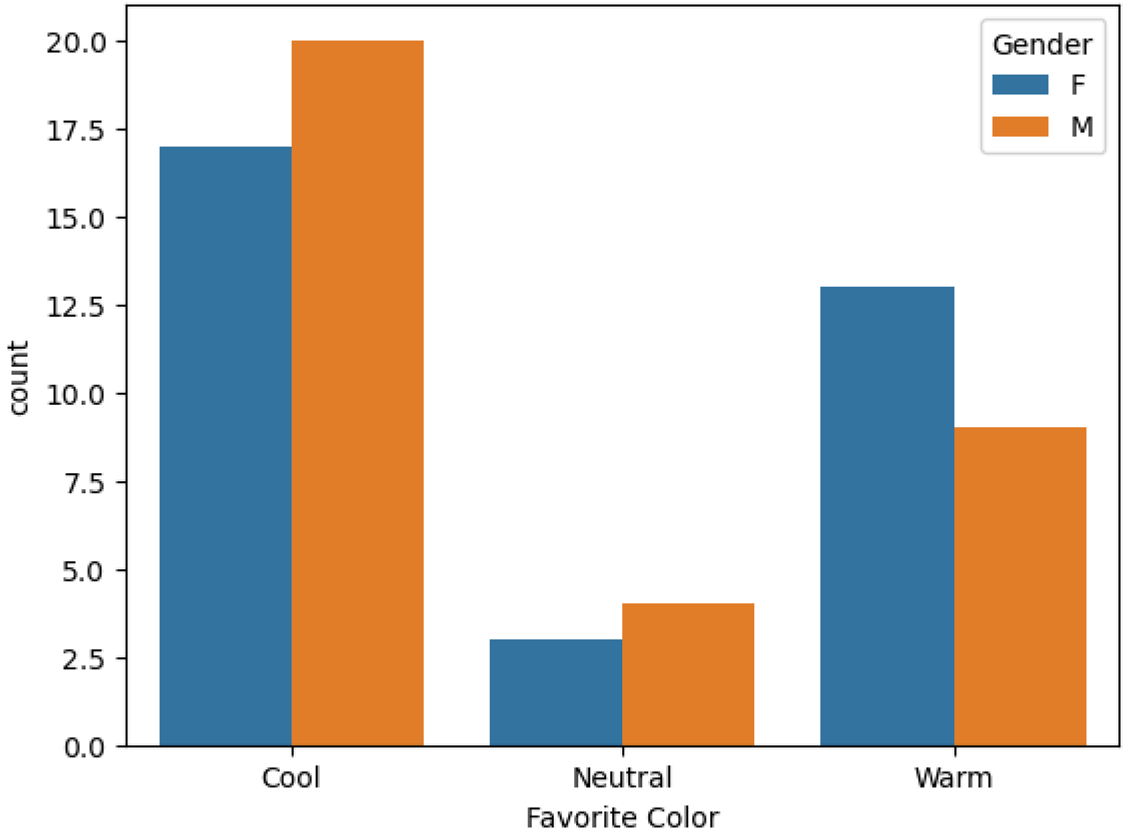


In [209…

```
sns.countplot(data = data , x = "Favorite Color" , hue = "Gender")
```

Out[209]:

```
<Axes: xlabel='Favorite Color', ylabel='count'>
```



In [210…

```
data["Favorite Music Genre"].value_counts()
```

Out[210]:

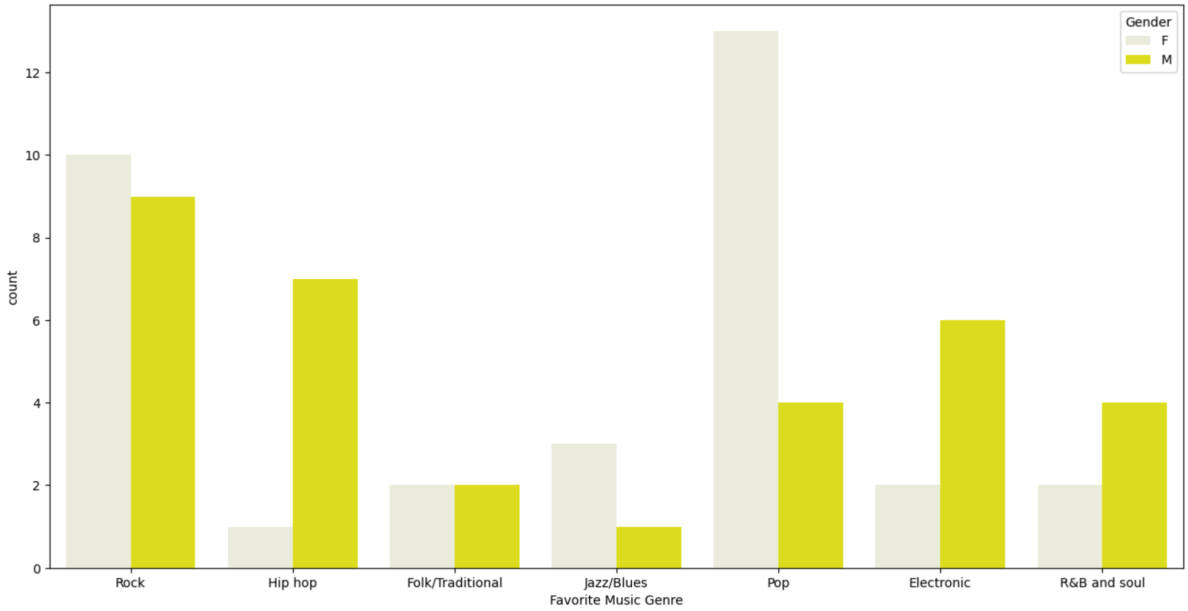
```
Favorite Music Genre
Rock                   19
Pop                    17
Hip hop                 8
Electronic              8
R&B and soul            6
Folk/Traditional        4
Jazz/Blues              4
Name: count, dtype: int64
```

In [211…

```
plt.figure(figsize = (16,8))
sns.countplot(data = data , x = "Favorite Music Genre" , hue = "Gender" , color = '
```

Out[211]:

```
<Axes: xlabel='Favorite Music Genre', ylabel='count'>
```



In [212…

```
data["Favorite Beverage"].value_counts()
```

Out[212]:

```
Favorite Beverage
Doesn't drink    14
Beer             13
Other            11
Wine             10
Vodka             9
Whiskey           9
Name: count, dtype: int64
```

In [213…

```
data["Favorite Soft Drink"].value_counts()
```

Out[213]:

```
Favorite Soft Drink
Coca Cola/Pepsi    32
Fanta              14
7UP/Sprite         13
Other               7
Name: count, dtype: int64
```

In [214…

```
data.head()
```

Out[214]:

	Favorite Color	Favorite Music Genre	Favorite Beverage	Favorite Soft Drink	Gender
0	Cool	Rock	Vodka	7UP/Sprite	F
1	Neutral	Hip hop	Vodka	Coca Cola/Pepsi	F
2	Warm	Rock	Wine	Coca Cola/Pepsi	F
3	Warm	Folk/Traditional	Whiskey	Fanta	F
4	Cool	Rock	Vodka	Coca Cola/Pepsi	F

In [215…

```
Favorite_Color = {"Cool":1,"Warm":2,"Neutral":3}
data["Favorite Color"] = data["Favorite Color"].map(Favorite_Color)
```

In [216…

```
data["Favorite Music Genre"].value_counts()
```

Out[216]:

```
Favorite Music Genre
Rock                19
Pop                 17
Hip hop              8
Electronic           8
R&B and soul         6
Folk/Traditional     4
Jazz/Blues           4
Name: count, dtype: int64
```

In [217…

```
Favorite_Music_Genre = {"Rock":1,"Pop":2,"Hip hop":3,"Electronic":4,"R&B and soul":
data["Favorite Music Genre"] = data["Favorite Music Genre"].map(Favorite_Music_Genr
```

In [218…

```
data["Favorite Beverage"].value_counts()
```

Out[218]:

```
Favorite Beverage
Doesn't drink    14
Beer             13
Other            11
Wine             10
Vodka             9
Whiskey           9
Name: count, dtype: int64
```

In [219…

```
Favorite_Beverage = {"Doesn't drink":1,"Beer":2,"Other":3,"Wine":4,"Vodka":5,"Whisk
data["Favorite Beverage"] = data["Favorite Beverage"].map(Favorite_Beverage)
```

In [ ]:

In [220…

```
data["Favorite Soft Drink"].value_counts()
```

Out[220]:

```
Favorite Soft Drink
Coca Cola/Pepsi    32
Fanta              14
7UP/Sprite         13
Other               7
Name: count, dtype: int64
```

In [221…

```
Favorite_soft_Drink = {"Coca Cola/Pepsi":1,"Fanta":2,"7UP/Sprite":3,"Other":4}
data["Favorite Soft Drink"] = data["Favorite Soft Drink"].map(Favorite_soft_Drink)
```

In [222…

```
data
```

Out[222]:

	Favorite Color	Favorite Music Genre	Favorite Beverage	Favorite Soft Drink	Gender
0	1	1	5	3	F
1	3	3	5	1	F
2	2	1	4	1	F
3	2	6	6	2	F
4	1	1	5	1	F
...	...	...	...	...	...
61	1	1	5	1	M
62	1	3	2	1	M
63	3	3	1	2	M
64	1	1	4	1	M
65	1	4	2	1	M

66 rows × 5 columns

In [223…

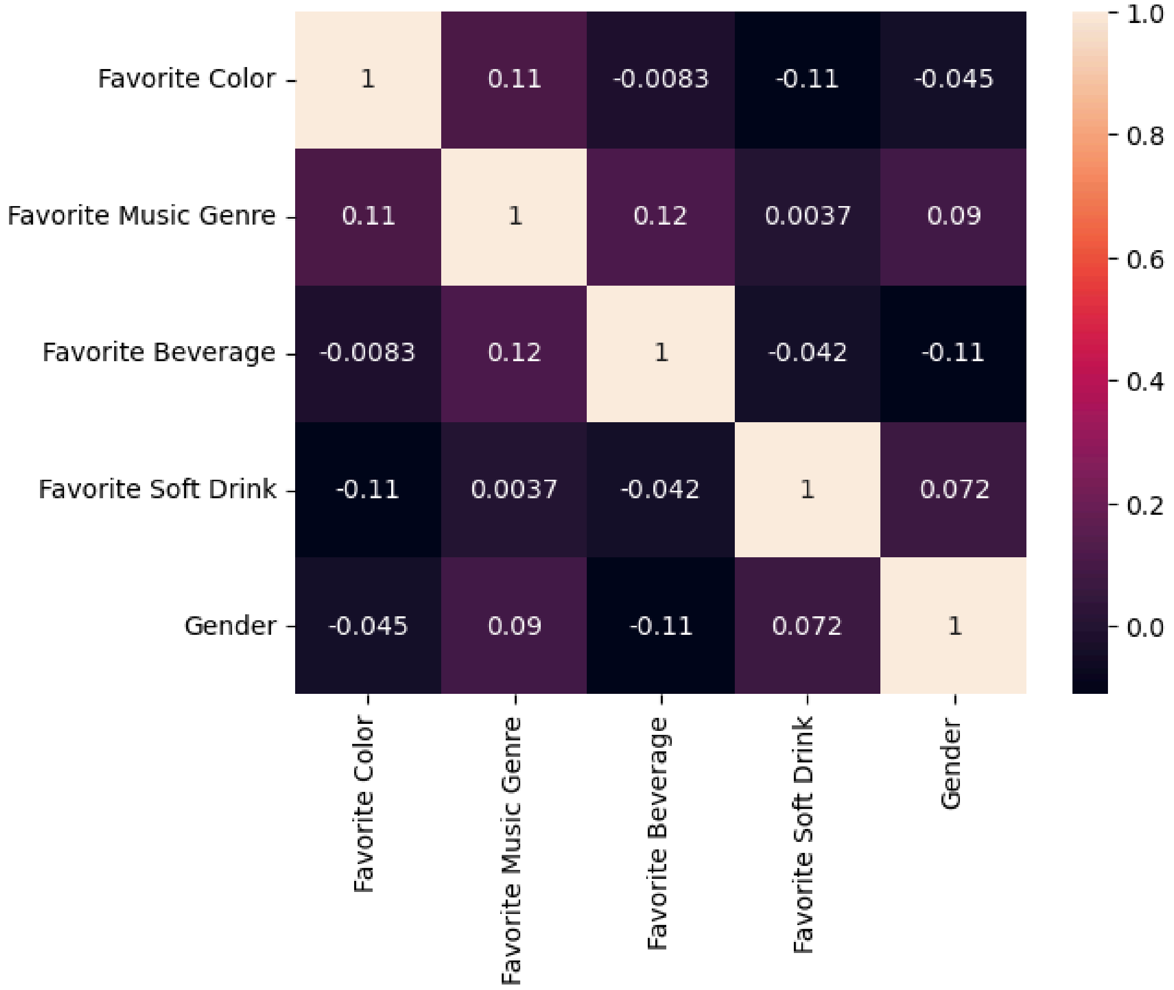
```
Gender = {"M":1,"F":0}
data["Gender"] = data["Gender"].map(Gender)
```

In [248…

```
sns.heatmap(data.corr() ,annot= True)
```

Out[248]:

```
<Axes: >
```



In [224…

```
data.isna().sum()
```

Out[224]:

```
Favorite Color          0
Favorite Music Genre    0
Favorite Beverage       0
Favorite Soft Drink     0
Gender                  0
dtype: int64
```

In [225…

```
data = data
```

In [226…

```
x = data.drop("Gender" , axis = 1)
y = data[["Gender"]]
```

In [227…

```
x
```

Out[227]:

	Favorite Color	Favorite Music Genre	Favorite Beverage	Favorite Soft Drink
0	1	1	5	3
1	3	3	5	1
2	2	1	4	1
3	2	6	6	2
4	1	1	5	1
...	...	...	...	...
61	1	1	5	1
62	1	3	2	1
63	3	3	1	2
64	1	1	4	1
65	1	4	2	1

66 rows × 4 columns

In [228…

```
y
```

Out[228]:

	Gender
0	0
1	0
2	0
3	0
4	0
...	...
61	1
62	1
63	1
64	1
65	1

66 rows × 1 columns

In [229...

```
x_train , x_test , y_train , y_test = train_test_split(x,y,test_size = 0.2 , random
```

In [230...

```
model = LogisticRegression()
```

In [231...

```
model.fit(x_train,y_train)
```

Out[231]:

```
▾ LogisticRegression
LogisticRegression()
```

In [232...

```
prediction = model.predict(x_test)
```

In [233...

```
prediction[11]
```

Out[233]:

```
1
```

In [234...

```
y_test.iloc[11,:]
```

Out[234]:

```
Gender    1
Name: 50, dtype: int64
```

In [235...

```
error = mean_squared_error(prediction , y_test)
```

In [236...

```
error
```

Out[236]:

```
0.42857142857142855
```

In [237...

```
knn = KNeighborsClassifier(n_neighbors=5)
```

In [238...

```
knn.fit(x_train,y_train)
```

Out[238]:

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

In [239…

```
y_prediction = knn.predict(x_test)
```

In [240…

```
y_prediction
```

Out[240]:

```
array([0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1], dtype=int64)
```

In [241…

```
y_test
```

Out[241]:

	Gender
21	0
2	0
56	1
3	0
6	0
38	1
20	0
19	0
52	1
26	0
45	1
50	1
17	0
34	1

In [242…

```
error2 = mean_squared_error(y_prediction , y_test)
```

In [243…

```
error2
```

Out[243]:

```
0.42857142857142855
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: