Data Analytics Project On Online Foods

Online Foods Dataset

This dataset is derived from transactions on an online food ordering service, documenting various factors over a set period. It includes demographic details like age, gender, marital status, occupation, monthly income, education level, and family size. Additionally, it captures location specifics through coordinates and postal codes.

Online food platforms provide unparalleled convenience, allowing customers to browse menus, place orders, and arrange delivery or pickup with just a few taps or clicks. This accessibility is particularly valuable for busy individuals, families, and professionals who may not have the time or inclination to cook or dine out regularly.

Problem Statements

In this dataset we have to identify user demographics (such as age , gender, or marital status) correlate with their food preferences ?

1.How does age impact food preferences?

2.Are there gender-based differences in food choices?

3.Does marital status influence food ordering behavior?

Data Dictionary

In the fast-paced world of online food ordering, efficient management of data is essential for seamless operations and exceptional customer experiences. The Data Dictionary for Online Foods serves as a comprehensive reference guide outlining the structure and attributes of the dataset or database used in online food ordering systems.

This document aims to standardize terminology, define key data elements, and provide clarity on the information stored within the online food platform. By establishing a common language and structure, the data dictionary facilitates effective communication among stakeholders, including developers, analysts, and decision-makers. The data dictionary plays a crucial role in various aspects.Here some of the attributes on online foods dataset

Age: Age of the customer.

Gender: Gender of the customer.

Marital Status: Marital status of the customer.

Occupation: Occupation of the customer.

Monthly Income: Monthly income of the customer.

Educational Qualifications: Educational qualifications of the customer.

Family Size: Number of individuals in the customer's family.

Latitude: Latitude of the customer's location.

Longitude: Longitude of the customer's location.

Pin Code: Pin code of the customer's location.

Output: Current status of the order

Feedback: Feedback provided by the customer after receiving the order.

Data Preprocessing

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model.

We imported the Pandas library and loaded our dataset into a DataFrame named 'df'. Now we can start exploring our data using Pandas methods and Seaborn for visualization.

In [18]:
```
import pandas as pd
```

```
import seaborn as sns
df=pd.read_csv("onlinefoods.csv")
df.head()
```

Out[18]:

	Age	Gender	Marital Status	Occupation	Monthly Income	Educational Qualifications	Family size	latitude	longitude	Pin code	Output	Feedback	Unnamed: 12
0	20	Female	Single	Student	No Income	Post Graduate	4	12.9766	77.5993	560001	Yes	Positive	Yes
1	24	Female	Single	Student	Below Rs.10000	Graduate	3	12.9770	77.5773	560009	Yes	Positive	Yes
2	22	Male	Single	Student	Below Rs.10000	Post Graduate	3	12.9551	77.6593	560017	Yes	Negative	Yes
3	22	Female	Single	Student	No Income	Graduate	6	12.9473	77.5616	560019	Yes	Positive	Yes
4	22	Male	Single	Student	Below Rs.10000	Post Graduate	4	12.9850	77.5533	560010	Yes	Positive	Yes

The df.shape attribute returns a tuple representing the dimensions of the DataFrame, where the first element indicates the number of rows and the second element indicates the number of columns.

In [2]:
```
df.shape
```

Out[2]:
```
(388, 13)
```

The df.size attribute returns the total number of elements in the DataFrame, which is calculated by multiplying the number of rows by the number of columns. This attribute gives you the total count of data points in your DataFrame.

In [3]:
```
df.size
```

Out[3]:
```
5044
```

The df.info() method provides a concise summary of the DataFrame, including the column names, data types, and the number of non-null values in each column. It's useful for getting an overview of the dataset.

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 388 entries, 0 to 387
Data columns (total 13 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Age                          388 non-null    int64  
 1   Gender                       388 non-null    object 
 2   Marital Status               388 non-null    object 
 3   Occupation                   388 non-null    object 
 4   Monthly Income               388 non-null    object 
 5   Educational Qualifications   388 non-null    object 
 6   Family size                  388 non-null    int64  
 7   latitude                     388 non-null    float64
 8   longitude                    388 non-null    float64
 9   Pin code                     388 non-null    int64  
 10  Output                       388 non-null    object 
 11  Feedback                     388 non-null    object 
 12  Unnamed: 12                  388 non-null    object 
dtypes: float64(2), int64(3), object(8)
memory usage: 39.5+ KB
```

The df.describe() method generates descriptive statistics for numerical columns in the DataFrame. It provides information such as count, mean, standard deviation, minimum, quartiles, and maximum values. This method gives you a quick overview of the distribution and central tendency of numerical data in your dataset.

In [5]:
```
df.describe()
```

Out[5]:

	Age	Family size	latitude	longitude	Pin code
count	388.000000	388.000000	388.000000	388.000000	388.000000
mean	24.628866	3.280928	12.972058	77.600160	560040.113402
std	2.975593	1.351025	0.044489	0.051354	31.399609
min	18.000000	1.000000	12.865200	77.484200	560001.000000
25%	23.000000	2.000000	12.936900	77.565275	560010.750000
50%	24.000000	3.000000	12.977000	77.592100	560033.500000
75%	26.000000	4.000000	12.997025	77.630900	560068.000000
max	33.000000	6.000000	13.102000	77.758200	560109.000000

The df.count is a method typically used in Python's pandas library to count non-null values for each column or row in a DataFrame. It returns a Series with the count of non-null values for each column or row, depending on the axis specified.

In [8]:
```
df.count
```

Out[8]:
```
<bound method DataFrame.count of      Age  Gender Marital Status Occupation  Monthly Income  \
0     20  Female         Single    Student       No Income
1     24  Female         Single    Student  Below Rs.10000
2     22    Male         Single    Student  Below Rs.10000
3     22  Female         Single    Student       No Income
4     22    Male         Single    Student  Below Rs.10000
..   ...     ...            ...        ...             ...
383   23  Female         Single    Student       No Income
384   23  Female         Single    Student       No Income
385   22  Female         Single    Student       No Income
386   23    Male         Single    Student  Below Rs.10000
387   23    Male         Single    Student       No Income

    Educational Qualifications  Family size  latitude  longitude  Pin code  \
0                Post Graduate            4   12.9766    77.5993    560001
1                     Graduate            3   12.9770    77.5773    560009
2                Post Graduate            3   12.9551    77.6593    560017
3                     Graduate            6   12.9473    77.5616    560019
4                Post Graduate            4   12.9850    77.5533    560010
..                         ...          ...       ...        ...       ...
383              Post Graduate            2   12.9766    77.5993    560001
384              Post Graduate            4   12.9854    77.7081    560048
385              Post Graduate            5   12.9850    77.5533    560010
386              Post Graduate            2   12.9770    77.5773    560009
387              Post Graduate            5   12.8988    77.5764    560078

    Output   Feedback Unnamed: 12
0      Yes   Positive         Yes
1      Yes   Positive         Yes
2      Yes  Negative          Yes
3      Yes   Positive         Yes
4      Yes   Positive         Yes
..     ...        ...         ...
383    Yes   Positive         Yes
384    Yes   Positive         Yes
385    Yes   Positive         Yes
386    Yes   Positive         Yes
387    Yes   Positive         Yes

[388 rows x 13 columns]>
```

EDA-Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in data science projects. It involves studying and exploring datasets to understand their main characteristics, discover patterns, identify outliers, and establish relationships between variables.

Univariate

Univariate analysis is the simplest form of quantitative data analysis. It's used to describe, summarize, and find patterns in the data from a single variable.

Univariate Analysis on Numerical Data

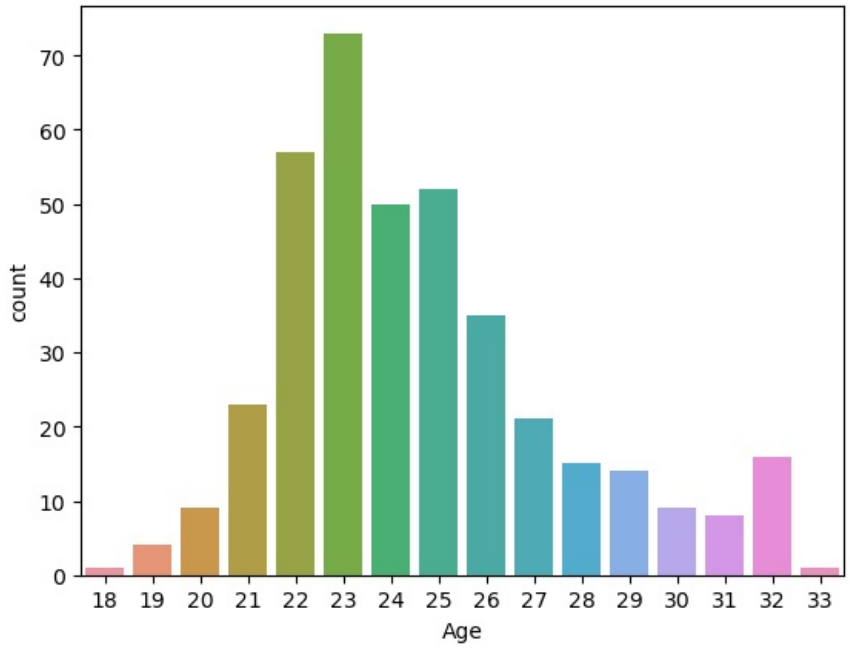
Now let's move to the analysis of this data. I will start by looking at the online food order based on the age of the customer

1.Countplot

The code you've provided seems to be using Seaborn, a Python data visualization library, to create a count plot based on the "Age" column of a DataFrame df. This plot would show the frequency of different age values in the dataset.

In [9]:
```
sns.countplot(data=df,x="Age")
```

Out[9]:
```
<Axes: xlabel='Age', ylabel='count'>
```



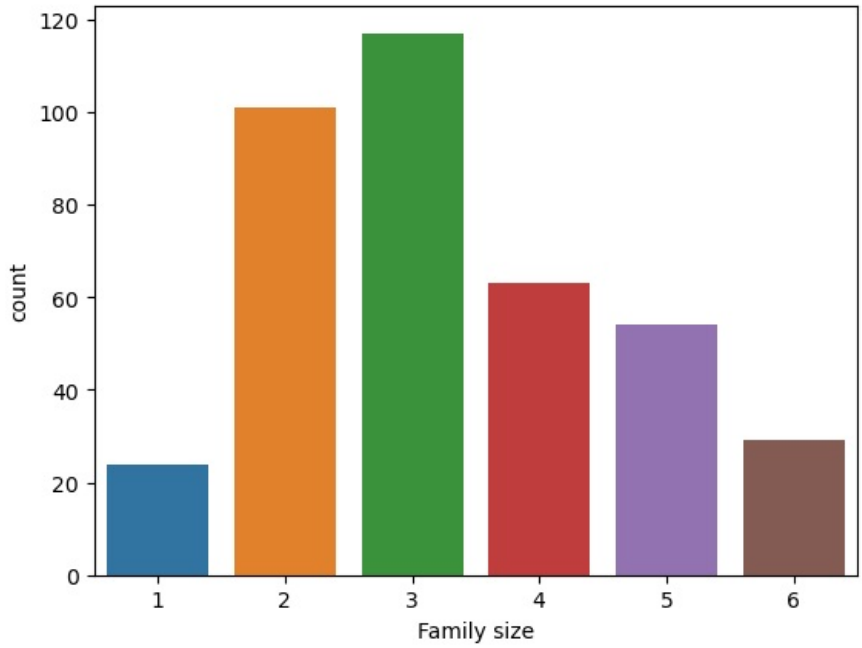
- We can see that the age group of 22-25 ordered the Online food. It also means this age group is the target of online food delivery companies.

Now let's have a look at the online food order decisions based on the size of the family of the customer

This code is attempting to create a count plot using Seaborn for the "Family size" column in the DataFrame df. A count plot will display the frequency of each unique value in the "Family size" column.

In [10]:
```
sns.countplot(data=df,x="Family size")
```

Out[10]:
```
<Axes: xlabel='Family size', ylabel='count'>
```



- Families with 2 and 3 members are ordering food often. These can be roommates, couples, or a family of three.

Univariate Analysis on Categorical Data

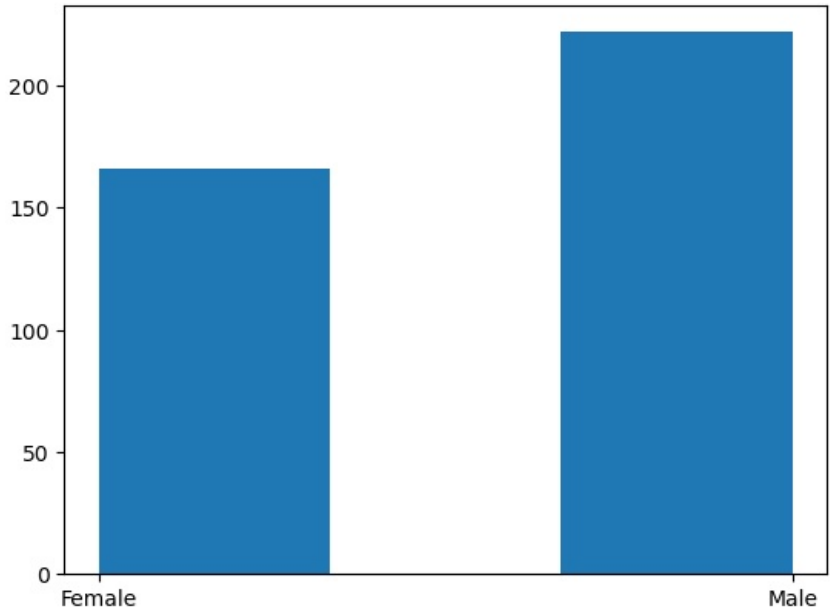
1.Histogram

Who Order food Online More : Male or Female

This code utilizes Matplotlib to create a histogram for the "Gender" column of the DataFrame df, with three bins

In [11]:
```
import matplotlib.pyplot as plt
plt.hist(df["Gender"],bins=3)
```

Out[11]:
```
(array([166.,   0., 222.]),
 array([0.        , 0.33333333, 0.66666667, 1.        ]),
 <BarContainer object of 3 artists>)
```



- According to the dataset, male customers are ordering more compared the females.

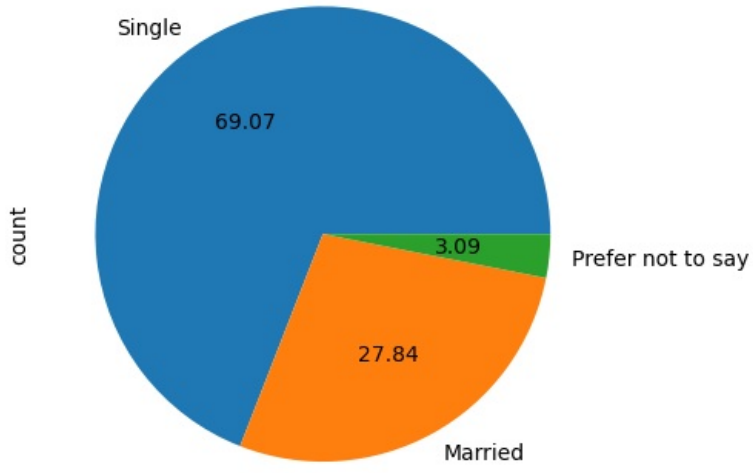
2.Pie Chart

Who order food Online More : Single or Married

This code generates a pie chart to visualize the distribution of values in the "Marital Status" column of the DataFrame df, using the value_counts() method to count the occurrences of each unique value.

In [12]:
```
df["Marital Status"].value_counts().plot(kind="pie",autopct="%.2f")
```

Out[12]:
```
<Axes: ylabel='count'>
```



- According to the above figure, 69.07% of the frequent customers are singles.

EDA-Bivariate

Bivariate analysis is one of the statistical analysis where two variables are observed. One variable here is dependent while the other is independent. These variables are usually denoted by X and Y.

Bivariate On Numerical - Categorical Data

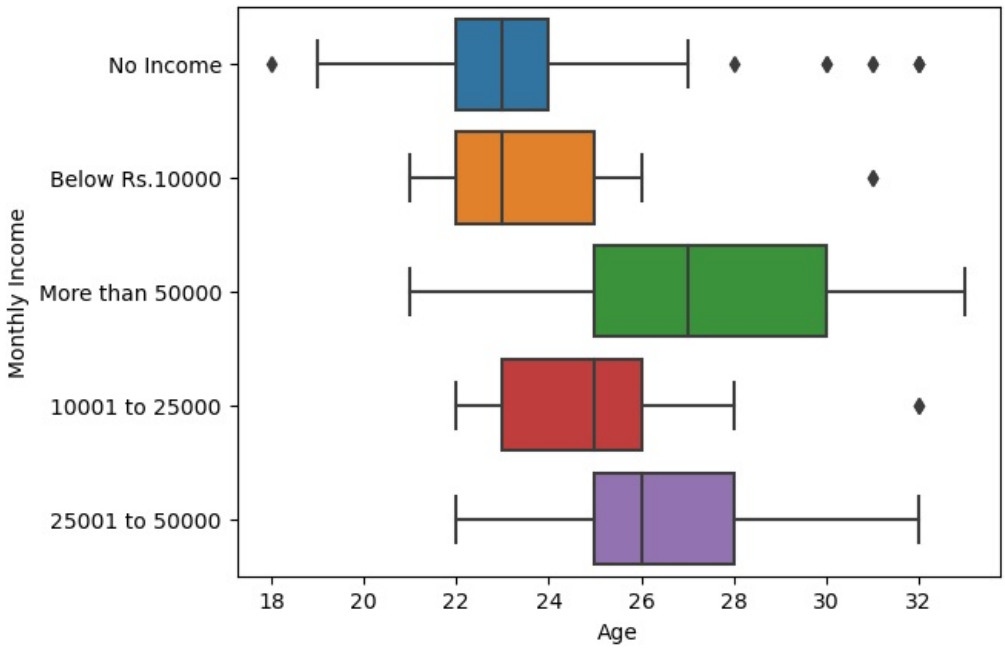
Now let's move to the Bivariate analysis of this data. I will start by looking at the online food order based on the age with the Monthly Income of the customer.

1.Box Plot

This code attempts to create a box plot using Seaborn to visualize the relationship between the "Age" and "Monthly Income" columns of the DataFrame df.

In [13]:
```
sns.boxplot(x=df['Age'],y=df['Monthly Income'])
```

Out[13]:
```
<Axes: xlabel='Age', ylabel='Monthly Income'>
```



- According to the Box plot ,More than 50000 Monthly Income customers are ordering more compared to others.

2.Scatter Plot

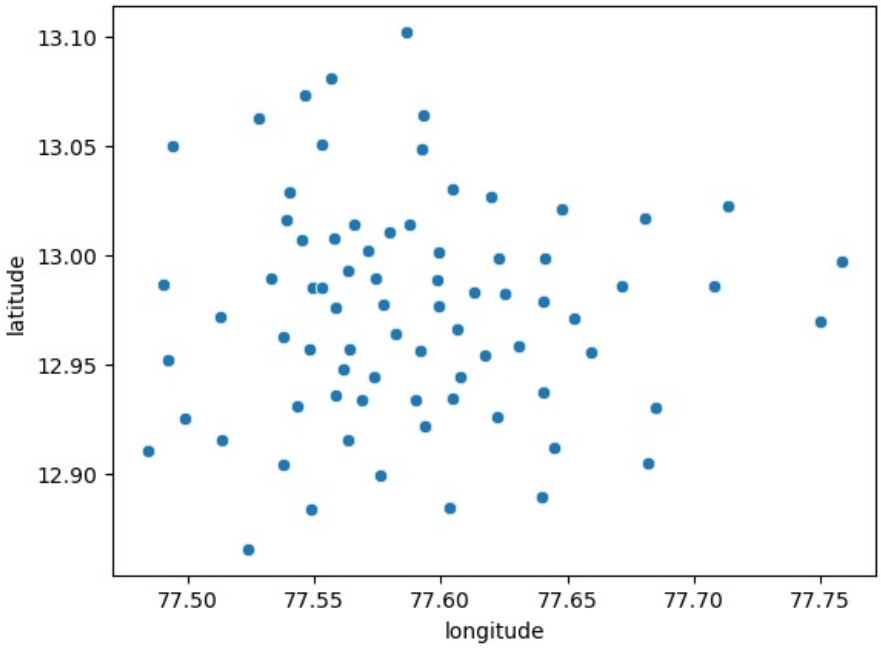
Latitude: Latitude specifies the angular distance of a location north or south of the Earth's equator. It provides geographic coordinates that can be used for spatial analysis and mapping purposes.

Longitude: Longitude denotes the angular distance of a location east or west of the Prime Meridian. Like latitude, it provides geographic coordinates for spatial analysis and mapping within the dataset.

The code we have provided uses Seaborn's scatterplot function to create a scatter plot, presumably to visualize geographical data represented by longitude and latitude coordinates.

In [14]:
```
sns.scatterplot(x=df['longitude'],y=df['latitude'])
```

Out[14]:
```
<Axes: xlabel='longitude', ylabel='latitude'>
```



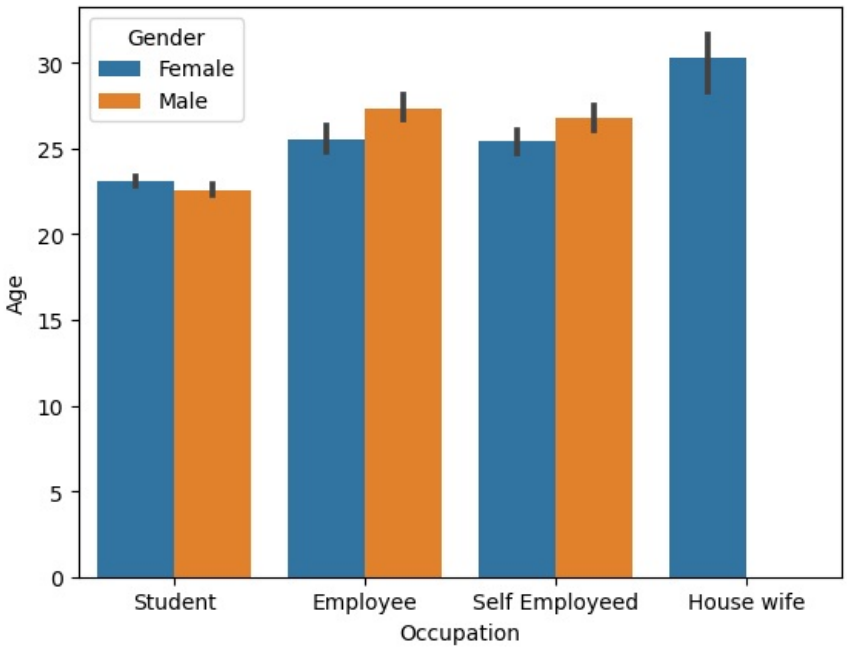
- According to the scatter plot,It shows the latitude and longitude of the customer where the order is placed.

3.Bar Plot

The code we provided utilizes Seaborn's barplot function to create a bar plot, likely to visualize the relationship between the "Occupation" and "Age" columns of the DataFrame df, with differentiation based on gender using the hue parameter.

In [15]:
```
sns.barplot(x=df['Occupation'],y=df['Age'],hue=df['Gender'])
```

Out[15]:
```
<Axes: xlabel='Occupation', ylabel='Age'>
```



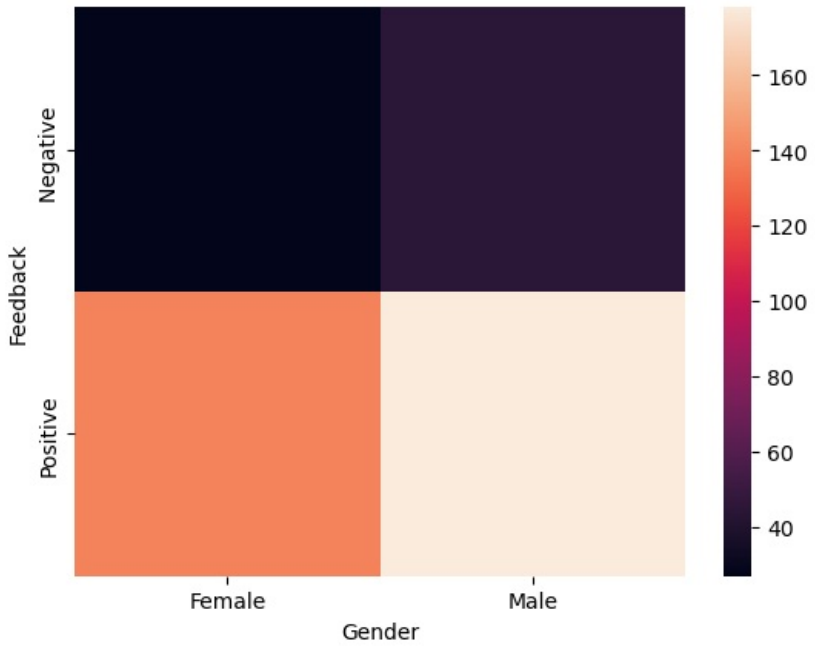
- According to the Bar plot , More than 20 years aged people are ordering more.Here we can also observe mens are dominating in Employee,Self Employeed bars and also female are dominating in Student,House Wife bars.

4.Heatmap

This code employs Seaborn's heatmap function to visualize a heatmap of the cross-tabulation between the "Feedback" and "Gender" columns of the DataFrame df, using pandas' crosstab function to compute the frequency table.

In [16]:
```
sns.heatmap(pd.crosstab(df['Feedback'],df['Gender']))
```

Out[16]:
```
<Axes: xlabel='Gender', ylabel='Feedback'>
```



- According to the Heatmap, The Males are given more positive and negative feedback compared to Females

Conclusion

In conclusion, age, gender, and marital status all play significant roles in shaping food preferences and ordering behavior, each influenced by a combination of physiological, social, and cultural factors.

1.We can observe that, The people at the age of 22-25 are prefers the online food mostly.

2.In this dataset we observe that, Males are ordering more as compared to Females.

3.Here we can identify that 69.07% of the frequent customers are singles.