In [5]:

```python
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

In [8]:

```python
df = pd.read_csv('onlinefoods.csv')
df
```

Out[8]:

	Age	Gender	Marital Status	Occupation	Monthly Income	Educational Qualifications	Family size	latitude	longitude
0	20	Female	Single	Student	No Income	Post Graduate	4	12.9766	77.5993
1	24	Female	Single	Student	Below Rs.10000	Graduate	3	12.9770	77.5773
2	22	Male	Single	Student	Below Rs.10000	Post Graduate	3	12.9551	77.6593
3	22	Female	Single	Student	No Income	Graduate	6	12.9473	77.5616
4	22	Male	Single	Student	Below Rs.10000	Post Graduate	4	12.9850	77.5533
...	...	...	...	...	...	...	...	...	...
383	23	Female	Single	Student	No Income	Post Graduate	2	12.9766	77.5993
384	23	Female	Single	Student	No Income	Post Graduate	4	12.9854	77.7081
385	22	Female	Single	Student	No Income	Post Graduate	5	12.9850	77.5533
386	23	Male	Single	Student	Below Rs.10000	Post Graduate	2	12.9770	77.5773
387	23	Male	Single	Student	No Income	Post Graduate	5	12.8988	77.5764

388 rows × 13 columns

In [10]:

```python
print(df.isnull().sum())
```

```
Age                           0
Gender                        0
Marital Status                0
Occupation                    0
Monthly Income                0
Educational Qualifications    0
Family size                   0
latitude                      0
longitude                     0
Pin code                      0
Output                        0
Feedback                      0
Unnamed: 12                   0
dtype: int64
```

In [11]:

```
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
```

In [13]:

```
cate_col=df.select_dtypes(include=['object']).columns
print(cate_col)
```

```
Index(['Gender', 'Marital Status', 'Occupation', 'Monthly Income',
       'Educational Qualifications', 'Output', 'Feedback', 'Unnamed: 12'],
      dtype='object')
```

In [15]:

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
for col in cate_col:
    df[col]=label_encoder.fit_transform(df[col])
df
```

Out[15]:

	Age	Gender	Marital Status	Occupation	Monthly Income	Educational Qualifications	Family size	latitude	longitude	
0	20	0	2	3	4	2	4	12.9766	77.5993	5
1	24	0	2	3	2	0	3	12.9770	77.5773	5
2	22	1	2	3	2	2	3	12.9551	77.6593	5
3	22	0	2	3	4	0	6	12.9473	77.5616	5
4	22	1	2	3	2	2	4	12.9850	77.5533	5
...	...	...	...	...	...	...	...	...	...	
383	23	0	2	3	4	2	2	12.9766	77.5993	5
384	23	0	2	3	4	2	4	12.9854	77.7081	5
385	22	0	2	3	4	2	5	12.9850	77.5533	5
386	23	1	2	3	2	2	2	12.9770	77.5773	5
387	23	1	2	3	4	2	5	12.8988	77.5764	5

388 rows × 13 columns

In [33]:

```
X=df.drop(columns=['Output','Feedback'])
y=df['Output']
print(X)
print(X)

```

```
     Age  Gender  Marital Status  Occupation  Monthly Income  \
0     20       0               2           3               4   
1     24       0               2           3               2   
2     22       1               2           3               2   
3     22       0               2           3               4   
4     22       1               2           3               2   
..   ...     ...             ...         ...             ...   
383   23       0               2           3               4   
384   23       0               2           3               4   
385   22       0               2           3               4   
386   23       1               2           3               2   
387   23       1               2           3               4   

     Educational Qualifications  Family size  latitude  longitude  Pin cod
e  \
0                             2            4   12.9766    77.5993    56000
1   
1                             0            3   12.9770    77.5773    56000
9   
2                             2            3   12.9551    77.6593    56001
7   
3                             0            6   12.9473    77.5616    56001
9   
4                             2            4   12.9850    77.5533    56001
0   
..                          ...          ...       ...        ...         
...   
383                           2            2   12.9766    77.5993    56000
1   
384                           2            4   12.9854    77.7081    56004
8   
385                           2            5   12.9850    77.5533    56001
0   
386                           2            2   12.9770    77.5773    56000
9   
387                           2            5   12.8988    77.5764    56007
8   

     Unnamed: 12  
0              1  
1              1  
2              1  
3              1  
4              1  
..           ...  
383            1  
384            1  
385            1  
386            1  
387            1  

[388 rows x 11 columns]
     Age  Gender  Marital Status  Occupation  Monthly Income  \
0     20       0               2           3               4   
1     24       0               2           3               2   
2     22       1               2           3               2   
3     22       0               2           3               4   
4     22       1               2           3               2   
..   ...     ...             ...         ...             ...   
383   23       0               2           3               4   
384   23       0               2           3               4   
```

```
385   22        0               2           3               4
386   23        1               2           3               2
387   23        1               2           3               4

     Educational Qualifications  Family size  latitude  longitude  Pin cod
e  \
0                             2            4   12.9766    77.5993    56000
1
1                             0            3   12.9770    77.5773    56000
9
2                             2            3   12.9551    77.6593    56001
7
3                             0            6   12.9473    77.5616    56001
9
4                             2            4   12.9850    77.5533    56001
0
..                          ...          ...       ...        ...
...
383                           2            2   12.9766    77.5993    56000
1
384                           2            4   12.9854    77.7081    56004
8
385                           2            5   12.9850    77.5533    56001
0
386                           2            2   12.9770    77.5773    56000
9
387                           2            5   12.8988    77.5764    56007
8

     Unnamed: 12
0              1
1              1
2              1
3              1
4              1
..           ...
383            1
384            1
385            1
386            1
387            1

[388 rows x 11 columns]
```

In [27]:

```python
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

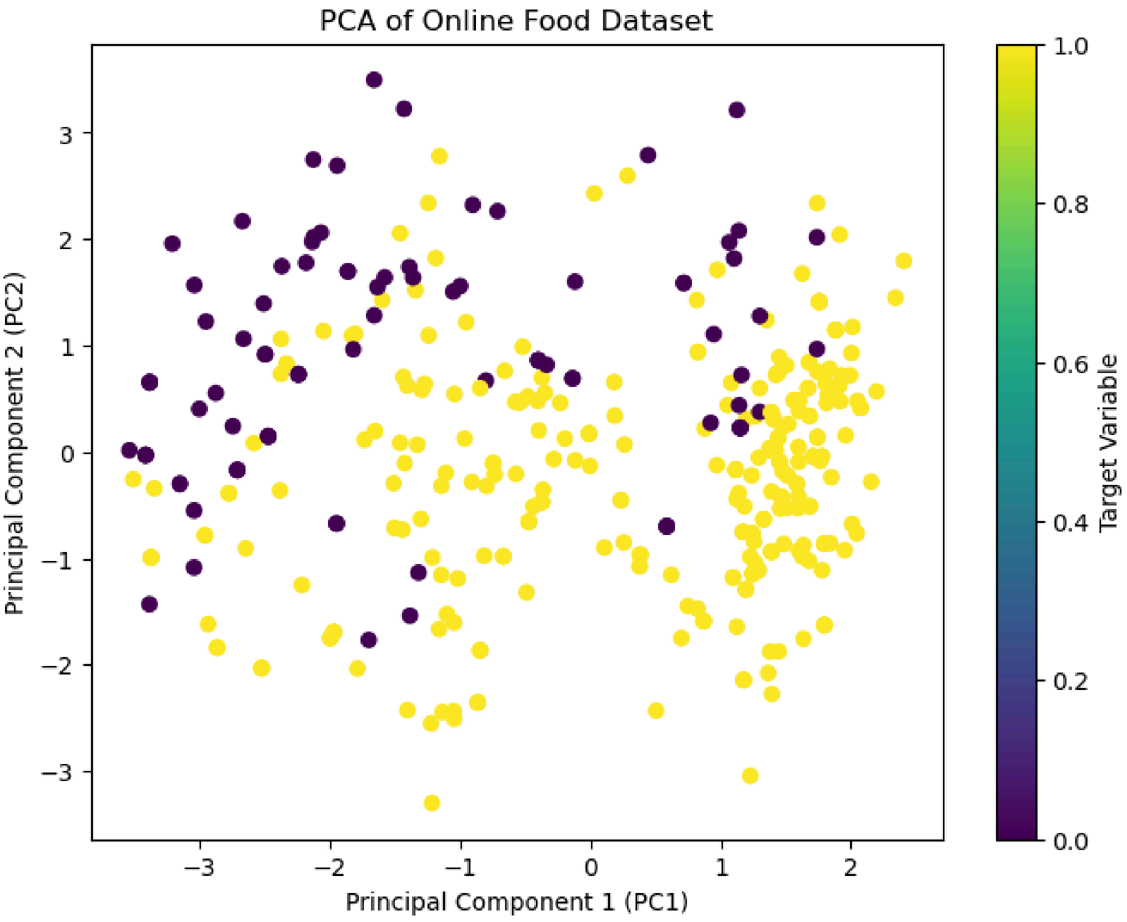
In [28]:

```
from sklearn.decomposition import PCA

# Initialize PCA (start with 2 components for visualization)
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Explained variance ratio
explained_variance = pca.explained_variance_ratio_
print("Explained Variance Ratio:", explained_variance)
```

```
Explained Variance Ratio: [0.25474049 0.12830153]
```

In [34]:

```
plt.figure(figsize=(8, 6))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y)
plt.xlabel('Principal Component 1 (PC1)')
plt.ylabel('Principal Component 2 (PC2)')
plt.title('PCA of Online Food Dataset')
plt.colorbar(label='Target Variable')
plt.show()
```



In [36]:

```python
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_s
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score


knn = KNeighborsClassifier(n_neighbors=5)

# Train the model
knn.fit(X_train, y_train)

# Make predictions
y_pred = knn.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy without PCA:", accuracy)
```

```
Accuracy without PCA: 0.9914529914529915
```

In [37]:

```python
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
knn_pca = KNeighborsClassifier(n_neighbors=5)


knn_pca.fit(X_train_pca, y_train)


y_pred_pca = knn_pca.predict(X_test_pca)


accuracy_pca = accuracy_score(y_test, y_pred_pca)
print("Accuracy with PCA:", accuracy_pca)
```

```
Accuracy with PCA: 0.8632478632478633
```

In [38]:

```python
print("Accuracy without PCA:", accuracy)
print("Accuracy with PCA:", accuracy_pca)
```

```
Accuracy without PCA: 0.9914529914529915
Accuracy with PCA: 0.8632478632478633
```

In [ ]:

```python

```