



# Online Food Order Prediction with Machine Learning



AMAN KHARWAL / ⌚ JUNE 7, 2022 / 📁 MACHINE LEARNING

There has been a high demand for online food orders after the introduction of Swiggy and Zomato in the market. Food delivery companies use your buying habits to make the delivery process faster. The food order prediction system is one of the useful techniques these companies can use to make the entire delivery process fast. In this article, I will take you through the task of Online Food Order Prediction with Machine Learning using Python.



**Uplevel uptime.  
Chillax, code, and ship.**

Start for Free



## Online Food Order Prediction: Use Case

After implementing the complete online food delivery system, companies like Swiggy and Zomato will always try to improve them. The main objective of these companies toward their customers is to deliver the food at the right time. To deliver the food faster, these companies identify areas where the demand for online food orders is high and employ more delivery partners in those locations. It helps deliver food faster in areas with more orders.

These companies have so much data about their customers that they now know the food ordering habits of all customers. With this data, they can also predict whether or not a customer will order again from their app. It is a good technique for identifying areas, families and customer types with more business opportunities.

I hope you now have understood the use case of online food order prediction systems. In the section below, I will take you through how you can train a Machine Learning model to predict online food orders from a particular customer.

## Online Food Order Prediction using Python

Now let's start with the task of online food order prediction with machine learning. **You can download the dataset I am using for this task from [here](#).** Let's start with importing the necessary Python libraries and the **dataset**:



```
1 import numpy as np
2 import pandas as pd
3 import numpy as np
4 import plotly.express as px
5 import plotly.graph_objects as go
6 import matplotlib.pyplot as plt
7 import seaborn as sns
8 sns.set_theme(style="whitegrid")
9
10 data = pd.read_csv("onlinefoods.csv")
```

```
11 print(data.head())
```

	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	

	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	

	Output	Feedback	Unnamed: 12
0	Yes	Positive	Yes
1	Yes	Positive	Yes
2	Yes	Negative	Yes
3	Yes	Positive	Yes
4	Yes	Positive	Yes

So the dataset contains information like:

1. the age of the customer
2. marital status of the customer
3. occupation of the customer
4. monthly income of the customer
5. educational qualification of the customer
6. family size of the customer
7. latitude and longitude of the location of the customer
8. pin code of the residence of the customer
9. did the customer order again (Output)
10. Feedback of the last order (Positive or Negative)

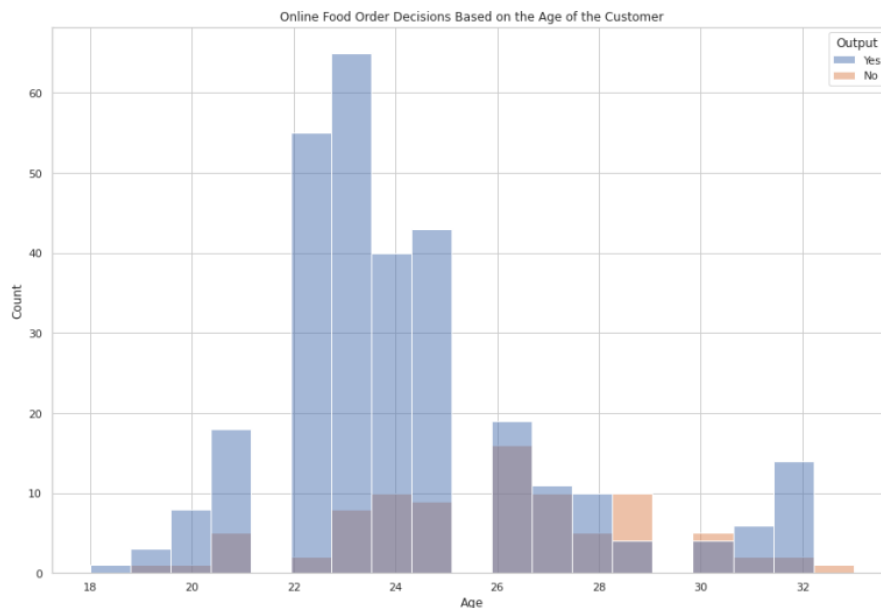
Let's have a look at the information about all the columns in the dataset:

```
1 print(data.info())
```

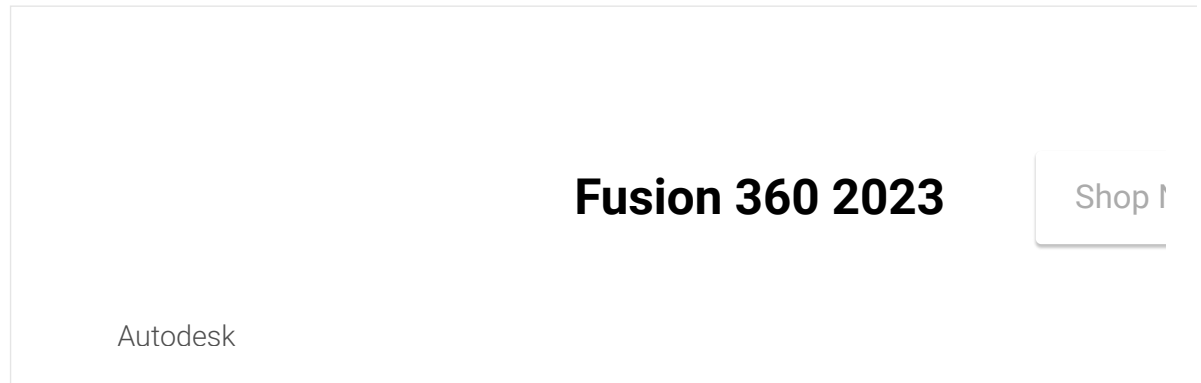
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 388 entries, 0 to 387
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  -
 0   Age                 388 non-null   int64
 1   Gender              388 non-null   object
 2   Marital Status      388 non-null   object
 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
None
```

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

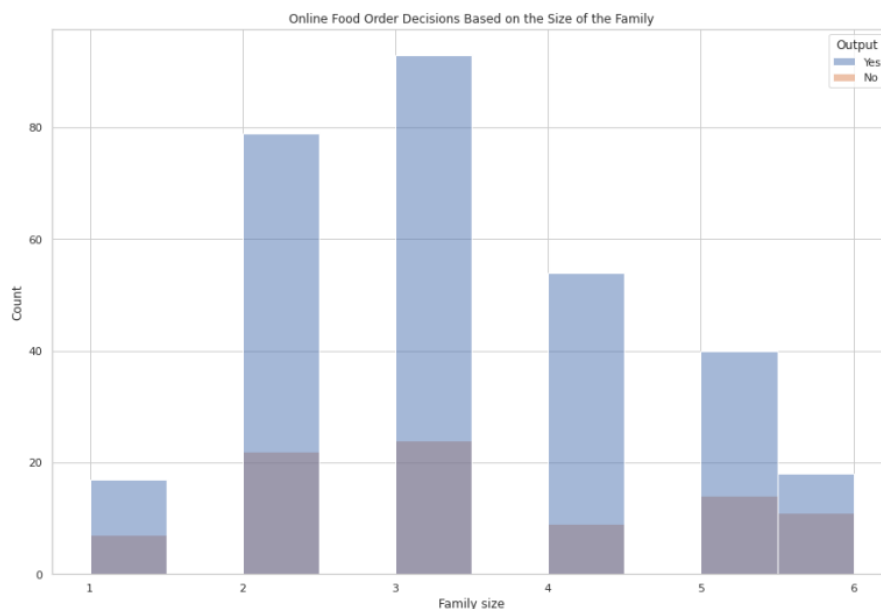
```
1 plt.figure(figsize=(15, 10))
2 plt.title("Online Food Order Decisions Based on the
3 sns.histplot(x="Age", hue="Output", data=data)
4 plt.show()
```



We can see that the age group of 22-25 ordered the food often again. 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:



```
1 plt.figure(figsize=(15, 10))
2 plt.title("Online Food Order Decisions Based on the
3 sns.histplot(x="Family size", hue="Output", data=da
4 plt.show()
```



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

Let's create a dataset of all the customers who ordered the food again:

```
1 buying_again_data = data.query("Output == 'Yes'")
2 print(buying_again_data.head())
```

	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

	Educational Qualifications	Family size	latitude	longitude	Pin
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

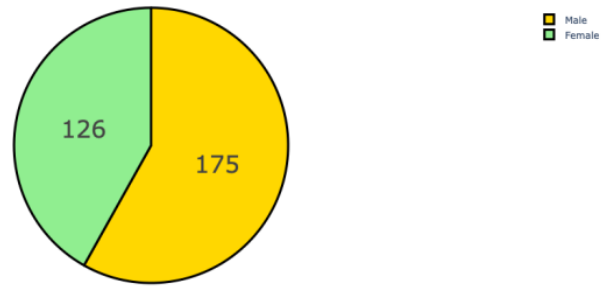
	Output	Feedback	Unnamed: 12
0	Yes	Positive	Yes
1	Yes	Positive	Yes
2	Yes	Negative	Yes
3	Yes	Positive	Yes
4	Yes	Positive	Yes

Now let's have a look at the gender column. Let's find who orders food more online:

```

1 gender = buying_again_data["Gender"].value_counts()
2 label = gender.index
3 counts = gender.values
4 colors = ['gold', 'lightgreen']
5
6 fig = go.Figure(data=[go.Pie(labels=label, values=c
7 fig.update_layout(title_text='Who Orders Food Onlin
8 fig.update_traces(hoverinfo='label+percent', textin
9                     marker=dict(colors=colors, line=d
10 fig.show()
```

Who Orders Food Online More: Male Vs. Female

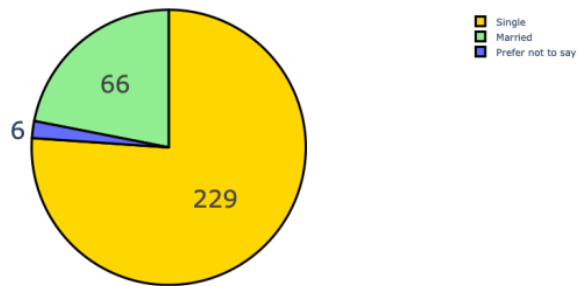


According to the dataset, male customers are ordering more compared to the females. Now let's have a look at the marital status of the customers who ordered again:

--	--

```
1 marital = buying_again_data["Marital Status"].value
2 label = marital.index
3 counts = marital.values
4 colors = ['gold', 'lightgreen']
5
6 fig = go.Figure(data=[go.Pie(labels=label, values=c
7 fig.update_layout(title_text='Who Orders Food Onlin
8 fig.update_traces(hoverinfo='label+percent', textin
9                     marker=dict(colors=colors, line=d
10 fig.show()
```

Who Orders Food Online More: Married Vs. Singles

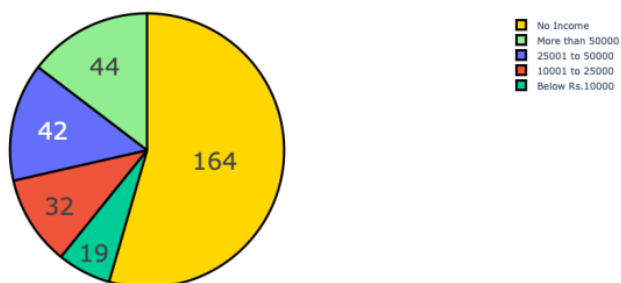


According to the above figure, 76.1% of the frequent customers are singles. Now let's have a look at what's the income group of the customers who ordered the food again:

```

1 income = buying_again_data["Monthly Income"].value_
2 label = income.index
3 counts = income.values
4 colors = ['gold', 'lightgreen']
5
6 fig = go.Figure(data=[go.Pie(labels=label, values=c
7 fig.update_layout(title_text='Which Income Group Or
8 fig.update_traces(hoverinfo='label+percent', textin
9                     marker=dict(colors=colors, line=d
10 fig.show()
```

Which Income Group Orders Food Online More



According to the above figure, 54% of the customers don't fall under any income group. They can be housewives or students.

Now let's prepare the data for the task of training a machine learning model. Here I will convert all the categorical features into numerical values:



```

1 data["Gender"] = data["Gender"].map({"Male": 1, "Fe
2 data["Marital Status"] = data["Marital Status"].map
3
4
5 data["Occupation"] = data["Occupation"].map({"Stude
6                               "Emplo
7                               "Self
8                               "House
9 data["Educational Qualifications"] = data["Educatio
10
11
12
13 data["Monthly Income"] = data["Monthly Income"].map
14
15
16
17
18 data["Feedback"] = data["Feedback"].map({"Positive"
19 print(data.head())

```

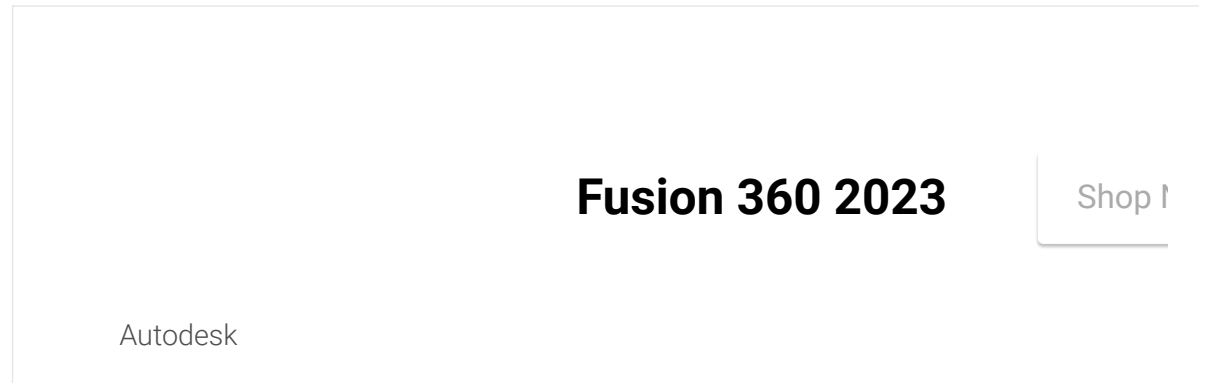
	Age	Gender	Marital Status	Occupation	Monthly Income \
0	20	0	1	1	0
1	24	0	1	1	10000
2	22	1	1	1	10000
3	22	0	1	1	0
4	22	1	1	1	10000

	Educational Qualifications	Family size	latitude	longitude	Pin
code \					
0	2	4	12.9766	77.5993	
560001					
1	1	3	12.9770	77.5773	
560009					
2	2	3	12.9551	77.6593	
560017					
3	1	6	12.9473	77.5616	
560019					
4	2	4	12.9850	77.5533	
560010					

	Output	Feedback	Unnamed: 12
0	Yes	1	Yes
1	Yes	1	Yes
2	Yes	0	Yes
3	Yes	1	Yes
4	Yes	1	Yes

# Online Food Order Prediction Model

Now let's train a machine learning model to predict whether a customer will order again or not. I will start by splitting the data into training and test sets:



```
1 #splitting data
2 from sklearn.model_selection import train_test_split
3 x = np.array(data[["Age", "Gender", "Marital Status",
4                   "Monthly Income", "Educational Q
5                   "Family size", "Pin code", "Feedb
6 y = np.array(data[["Output"]])
```

Now let's train the machine learning model:

```
1 # training a machine learning model
2 from sklearn.ensemble import RandomForestClassifier
3 xtrain, xtest, ytrain, ytest = train_test_split(x, y,
4                                                 test
5                                                 rand
6 model = RandomForestClassifier()
7 model.fit(xtrain, ytrain)
8 print(model.score(xtest, ytest))
```

```
0.9487179487179487
```

Now let's prepare a form to input the data of the customer and predict whether the customer will order the food again or not:

```
1 print("Enter Customer Details to Predict If the Cus
2 a = int(input("Enter the Age of the Customer: "))
3 b = int(input("Enter the Gender of the Customer (1
4 c = int(input("Marital Status of the Customer (1 =
```

```

5 d = int(input("Occupation of the Customer (Student
6 e = int(input("Monthly Income: "))
7 f = int(input("Educational Qualification (Graduate
8 g = int(input("Family Size: "))
9 h = int(input("Pin Code: "))
10 i = int(input("Review of the Last Order (1 = Positi
11 features = np.array([[a, b, c, d, e, f, g, h, i]])
12 print("Finding if the customer will order again: ",

```

```

Enter Customer Details to Predict If the Customer Will Order Again
Enter the Age of the Customer: 22
Enter the Gender of the Customer (1 = Male, 0 = Female): 1
Marital Status of the Customer (1 = Single, 2 = Married, 3 = Not
Revealed): 1
Occupation of the Customer (Student = 1, Employee = 2, Self Employeed
= 3, House wife = 4): 1
Monthly Income: 0
Educational Qualification (Graduate = 1, Post Graduate = 2, Ph.D = 3,
School = 4, Uneducated = 5): 1
Family Size: 3
Pin Code: 560010
Review of the Last Order (1 = Positive, 0 = Negative): 1

Finding if the customer will order again: ['Yes']

```

This is how you can train a machine learning model to predict online food orders.

## Summary

So this is how you can predict whether a customer will order food online again or not. The food order prediction system is one of the useful techniques food delivery companies can use to make the entire delivery process fast. I hope you liked this article on Online Food Delivery Prediction with Machine Learning. Feel free to ask valuable questions in the comments section below.

## Fusion 360 2023

[Shop Fusion 360](#)

Autodesk



Aman Kharwal

I'm a writer and data scientist on a mission to educate others about the incredible power of data 📊.

ARTICLES: 1354



PREVIOUS

NEXT



### Recommended For You

**Food Delivery  
Time  
Prediction  
using Python**



## Food Delivery Time Prediction using Python

January 2, 2023

# Job Recommendation System using Python



## Job Recommendation System using Python

December 12, 2022

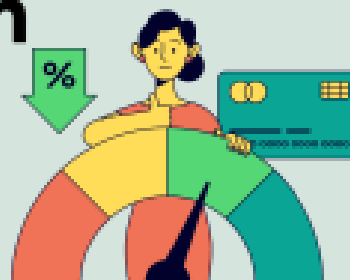
# Machine Learning Roadmap with Learning Resources



## Machine Learning Roadmap

December 6, 2022 / 2 Comments

# Credit Score Classification with Machine Learning



## Credit Score Classification with Machine Learning

December 5, 2022

### Leave a Reply

Enter your comment here...

 FACEBOOK  INSTAGRAM  MEDIUM  LINKEDIN

Copyright © Thecleverprogrammer.com 2023