

Waiter Tips Prediction using Python.

Now let's start the task of waiter tips analysis and prediction by importing the necessary Python libraries and the dataset:

Importing Libraries

```
In [1]: import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
```

Loading the Dataset

```
In [2]: tips_data = pd.read_csv("tips.csv")
print(tips_data.head())
```

| | total_bill | tip | sex | smoker | day | time | size |
|---|------------|------|--------|--------|-----|--------|------|
| 0 | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| 1 | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| 2 | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| 3 | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| 4 | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |

Below is the complete description of this dataset:

total_bill: Total bill in dollars including tax

tip: Tip given to waiter in dollars

sex: gender of the person paying the bill

smoker: whether the person smoked or not

day: day of the week

time: lunch or dinner

size: number of people

Now, let's move forward by analysing all the aspects affecting waiter tips:

Waiter Tips Analysis

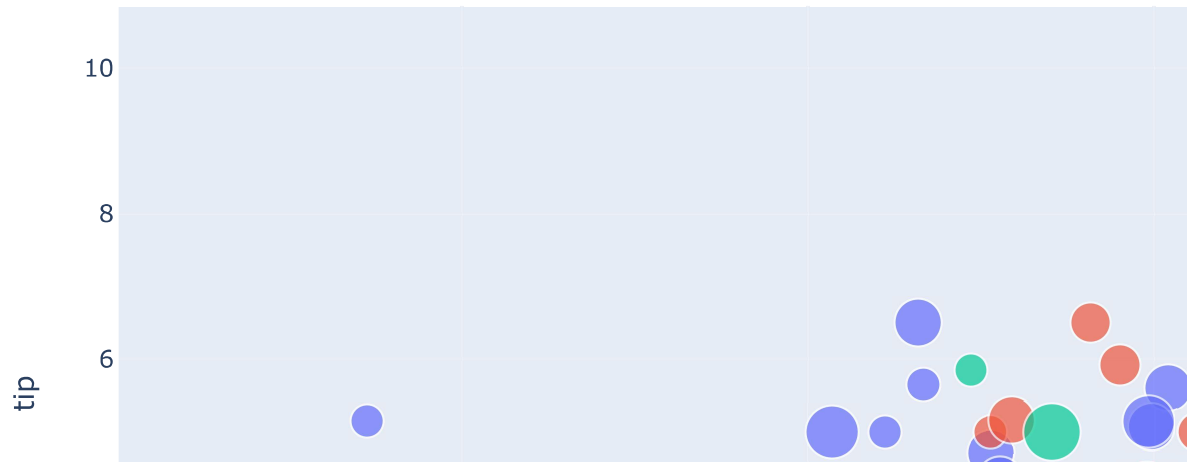
Let's have a look at the tips given to the waiters according to:

the total bill paid

number of people at a table

and the day of the week:

```
In [3]: figure = px.scatter(data_frame = tips_data, x = "total_bill", y = "tip", size = "size",  
figure.show()
```



Now let's have a look at the tips given to the waiters according to:

the total bill paid

the number of people at a table

and the gender of the person paying the bill:

```
In [4]: figure = px.scatter(data_frame = tips_data, x="total_bill",  
y="tip", size="size", color= "sex", trendline="ols")  
figure.show()
```



Now let's have a look at the tips given to the waiters according to:

the total bill paid

the number of people at a table

and the time of the meal:

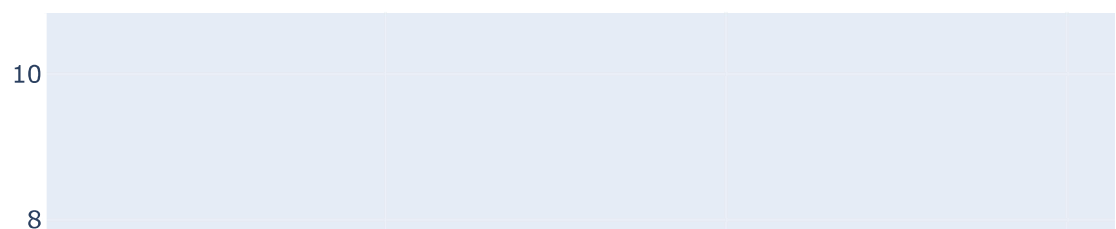
Now let's have a look at the tips given to the waiters according to:

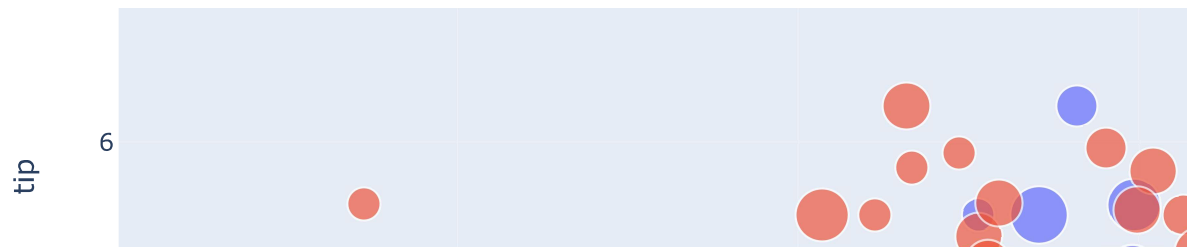
the total bill paid

the number of people at a table

and the time of the meal:

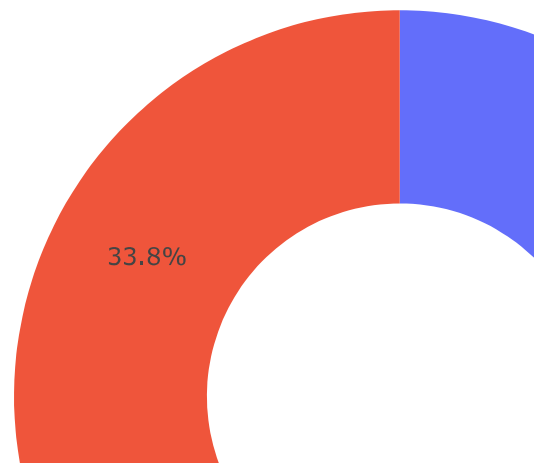
```
In [5]: figure = px.scatter(data_frame = tips_data, x="total_bill",
                             y="tip", size="size", color= "sex", trendline="ols")
figure.show()
```





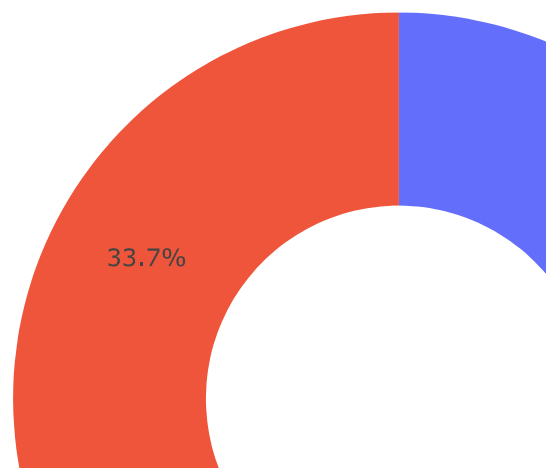
Now let's see the tips given to the waiters according to the days to find out which day the most tips are given to the waiters:

```
In [6]: figure = px.pie(tips_data,  
                        values='tip',  
                        names='day',hole = 0.5)  
figure.show()
```



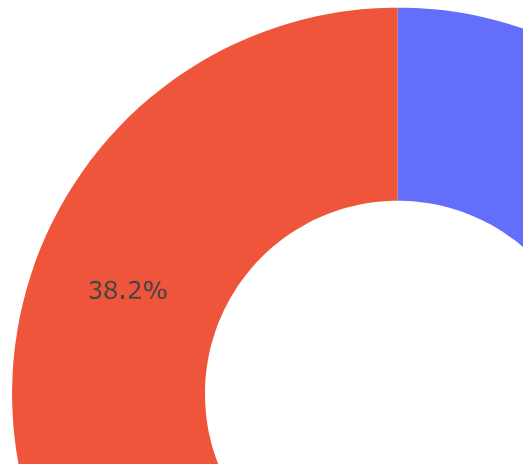
According to the visualization above, on Saturdays, most tips are given to the waiters. Now let's look at the number of tips given to waiters by gender of the person paying the bill to see who tips waiters the most:

```
In [8]: figure = px.pie(tips_data,  
                        values='tip',  
                        names='sex',hole = 0.5)  
figure.show()
```



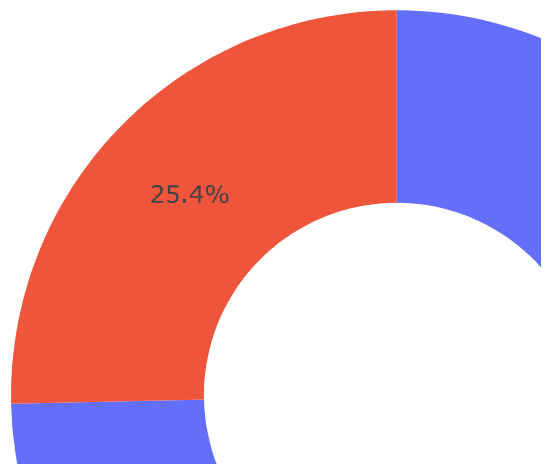
According to the visualization above, most tips are given by men. Now let's see if a smoker tips more or a non-smoker:

```
In [9]: figure = px.pie(tips_data,  
                        values='tip',  
                        names='smoker',hole = 0.5)  
figure.show()
```



According to the visualization above, non-smoker tips waiters more than smokers. Now let's see if most tips are given during lunch or dinner:

```
In [10]: figure = px.pie(tips_data,
                        values='tip',
                        names='time',hole = 0.5)
figure.show()
```



According to the visualization above, a waiter is tipped more during dinner.

So this is how we can analyze all the factors affecting waiter tips. Now in the section below, I will take you through how to train a machine learning model for the task of waiter tips prediction.

Waiter Tips Prediction Model

Before training a waiter tips prediction model, I will do some data transformation by transforming the categorical values into numerical values:

```
In [12]: tips_data["sex"] = tips_data["sex"].map({"Female": 0, "Male": 1})
tips_data["smoker"] = tips_data["smoker"].map({"No": 0, "Yes": 1})
tips_data["day"] = tips_data["day"].map({"Thur": 0, "Fri": 1, "Sat": 2, "Sun": 3})
tips_data["time"] = tips_data["time"].map({"Lunch": 0, "Dinner": 1})
tips_data.head()
```

```
Out[12]:
```

| | total_bill | tip | sex | smoker | day | time | size |
|---|------------|------|-----|--------|-----|------|------|
| 0 | 16.99 | 1.01 | 0 | 0 | 3 | 1 | 2 |
| 1 | 10.34 | 1.66 | 1 | 0 | 3 | 1 | 3 |
| 2 | 21.01 | 3.50 | 1 | 0 | 3 | 1 | 3 |
| 3 | 23.68 | 3.31 | 1 | 0 | 3 | 1 | 2 |
| 4 | 24.59 | 3.61 | 0 | 0 | 3 | 1 | 4 |

Now I will split the data into training and test sets:

Now below is how we can train a machine learning model for the task of waiter tips prediction using Python:

```
In [20]: # splitting dataset / columns / features into independent (Xi) and dependent(y) variable
import numpy as np
X = tips_data.drop(columns='tip')
X
```

```
Out[20]:
```

| | total_bill | sex | smoker | day | time | size |
|---|------------|-----|--------|-----|------|------|
| 0 | 16.99 | 0 | 0 | 3 | 1 | 2 |

| | total_bill | sex | smoker | day | time | size |
|-----|------------|-----|--------|-----|------|------|
| 1 | 10.34 | 1 | 0 | 3 | 1 | 3 |
| 2 | 21.01 | 1 | 0 | 3 | 1 | 3 |
| 3 | 23.68 | 1 | 0 | 3 | 1 | 2 |
| 4 | 24.59 | 0 | 0 | 3 | 1 | 4 |
| ... | ... | ... | ... | ... | ... | ... |
| 239 | 29.03 | 1 | 0 | 2 | 1 | 3 |
| 240 | 27.18 | 0 | 1 | 2 | 1 | 2 |
| 241 | 22.67 | 1 | 1 | 2 | 1 | 2 |
| 242 | 17.82 | 1 | 0 | 2 | 1 | 2 |
| 243 | 18.78 | 0 | 0 | 0 | 1 | 2 |

244 rows × 6 columns

```
In [22]: y = tips_data['tip']
y
```

```
Out[22]: 0      1.01
1      1.66
2      3.50
3      3.31
4      3.61
...
239    5.92
240    2.00
241    2.00
242    1.75
243    3.00
Name: tip, Length: 244, dtype: float64
```

```
In [ ]: from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(xtrain, ytrain)
```

```
In [24]: # splitting the data into training and test sets:

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,
                                                test_size=0.2, #20% is reserved for "t
                                                random_state=355)
```

Training & Evaluating the Linear Regression (ML) model

```
In [25]: from sklearn.linear_model import LinearRegression
```



```
model = LinearRegression()
model.fit(X_train,y_train)
```

Out[25]: LinearRegression()

```
In [26]: # calculating the Accuracy
model.score(X_test,y_test)
```

Out[26]: 0.3792579793220321

```
In [28]: X_test.head()
```

Out[28]:

| | total_bill | sex | smoker | day | time | size |
|------------|------------|-----|--------|-----|------|------|
| 96 | 27.28 | 1 | 1 | 1 | 1 | 2 |
| 132 | 11.17 | 0 | 0 | 0 | 0 | 2 |
| 121 | 13.42 | 0 | 0 | 0 | 0 | 2 |
| 143 | 27.05 | 0 | 0 | 0 | 0 | 6 |
| 107 | 25.21 | 1 | 1 | 2 | 1 | 2 |

Now let's test the performance of this model by giving inputs to this model according to the features that we have used to train this model:

```
In [29]: features = np.array([[27.28, 0, 0, 4, 1,2]])
model.predict(features)
```

Out[29]: array([3.93488281])

```
In [30]: # 3.62 is close to 4.00

y_test.head()
```

Out[30]:

| | |
|-----|------|
| 96 | 4.00 |
| 132 | 1.50 |
| 121 | 1.68 |
| 143 | 5.00 |
| 107 | 4.29 |

Name: tip, dtype: float64

Summary

So this is how you can predict waiter tips with machine learning using Python. Waiter Tips analysis is one of the popular data science case studies where we need to predict the tips given to a waiter for serving the food in a restaurant.