Uber_Data_Analysis

June 25, 2023

1 Introduction

Uber's data analysis plays a crucial role in understanding rider behavior, driver performance, market trends, and operational efficiency. By leveraging various datasets, Uber can gain valuable insights into customer preferences, pricing strategies, traffic patterns, and more. This enables Uber to enhance its services, ensure customer satisfaction, and streamline operations for maximum efficiency. Purpose of Uber Data Analysis:

The primary purpose of data analysis at Uber is to extract meaningful insights from the vast amount of data generated by its platform. These insights are then used to drive strategic decision-making, improve service offerings, and enhance the overall user experience.

[107]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import preprocessing
```

[108]:
```
uber=pd.read_csv("rideshare_kaggle.csv")
```

[109]:
```
uber.head(5)
```

[109]:

	id	timestamp	hour	day	month	\
0	424553bb-7174-41ea-aeb4-fe06d4f4b9d7	1.544953e+09	9	16	12	
1	4bd23055-6827-41c6-b23b-3c491f24e74d	1.543284e+09	2	27	11	
2	981a3613-77af-4620-a42a-0c0866077d1e	1.543367e+09	1	28	11	
3	c2d88af2-d278-4bfd-a8d0-29ca77cc5512	1.543554e+09	4	30	11	
4	e0126e1f-8ca9-4f2e-82b3-50505a09db9a	1.543463e+09	3	29	11	

```
              datetime          timezone            source    destination  \
0  2018-12-16 09:30:07  America/New_York  Haymarket Square  North Station
1  2018-11-27 02:00:23  America/New_York  Haymarket Square  North Station
2  2018-11-28 01:00:22  America/New_York  Haymarket Square  North Station
3  2018-11-30 04:53:02  America/New_York  Haymarket Square  North Station
4  2018-11-29 03:49:20  America/New_York  Haymarket Square  North Station

  cab_type  … precipIntensityMax uvIndexTime  temperatureMin  \
0     Lyft  …              0.1276  1544979600           39.89
1     Lyft  …              0.1300  1543251600           40.49
2     Lyft  …              0.1064  1543338000           35.36
3     Lyft  …              0.0000  1543507200           34.67
4     Lyft  …              0.0001  1543420800           33.10

   temperatureMinTime  temperatureMax  temperatureMaxTime  \
0          1545012000           43.68          1544968800
1          1543233600           47.30          1543251600
2          1543377600           47.55          1543320000
3          1543550400           45.03          1543510800
4          1543402800           42.18          1543420800

   apparentTemperatureMin  apparentTemperatureMinTime  apparentTemperatureMax  \
0                   33.73                  1545012000                   38.07
1                   36.20                  1543291200                   43.92
2                   31.04                  1543377600                   44.12
3                   30.30                  1543550400                   38.53
4                   29.11                  1543392000                   35.75

  apparentTemperatureMaxTime
0                 1544958000
1                 1543251600
2                 1543320000
3                 1543510800
4                 1543420800

[5 rows x 57 columns]
```

[110]:
```
uber.shape
```

[110]:
```
(693071, 57)
```

[111]:
```
uber['name'].value_counts()
```

[111]:
```
UberXL          55096
WAV             55096
Black SUV       55096
Black           55095
```

```
Taxi            55095
UberX           55094
UberPool        55091
Lux             51235
Lyft            51235
Lux Black XL    51235
Lyft XL         51235
Lux Black       51235
Shared          51233
Name: name, dtype: int64
```

[112]:
```
uber.columns
```

[112]:
```
Index(['id', 'timestamp', 'hour', 'day', 'month', 'datetime', 'timezone',
       'source', 'destination', 'cab_type', 'product_id', 'name', 'price',
       'distance', 'surge_multiplier', 'latitude', 'longitude', 'temperature',
       'apparentTemperature', 'short_summary', 'long_summary',
       'precipIntensity', 'precipProbability', 'humidity', 'windSpeed',
       'windGust', 'windGustTime', 'visibility', 'temperatureHigh',
       'temperatureHighTime', 'temperatureLow', 'temperatureLowTime',
       'apparentTemperatureHigh', 'apparentTemperatureHighTime',
       'apparentTemperatureLow', 'apparentTemperatureLowTime', 'icon',
       'dewPoint', 'pressure', 'windBearing', 'cloudCover', 'uvIndex',
       'visibility.1', 'ozone', 'sunriseTime', 'sunsetTime', 'moonPhase',
       'precipIntensityMax', 'uvIndexTime', 'temperatureMin',
       'temperatureMinTime', 'temperatureMax', 'temperatureMaxTime',
       'apparentTemperatureMin', 'apparentTemperatureMinTime',
       'apparentTemperatureMax', 'apparentTemperatureMaxTime'],
      dtype='object')
```

[113]:
```
uber.info
```

[113]:
```
<bound method DataFrame.info of                                      id
timestamp  hour  day  month  \
0       424553bb-7174-41ea-aeb4-fe06d4f4b9d7  1.544953e+09     9   16     12
1       4bd23055-6827-41c6-b23b-3c491f24e74d  1.543284e+09     2   27     11
2       981a3613-77af-4620-a42a-0c0866077d1e  1.543367e+09     1   28     11
3       c2d88af2-d278-4bfd-a8d0-29ca77cc5512  1.543554e+09     4   30     11
4       e0126e1f-8ca9-4f2e-82b3-50505a09db9a  1.543463e+09     3   29     11
…                                        …           …   …  …     …
693066  616d3611-1820-450a-9845-a9ff304a4842  1.543708e+09    23    1     12
693067  633a3fc3-1f86-4b9e-9d48-2b7132112341  1.543708e+09    23    1     12
693068  64d451d0-639f-47a4-9b7c-6fd92fbd264f  1.543708e+09    23    1     12
693069  727e5f07-a96b-4ad1-a2c7-9abc3ad55b4e  1.543708e+09    23    1     12
693070  e7fdc087-fe86-40a5-a3c3-3b2a8badcbda  1.543708e+09    23    1     12

                   datetime          timezone            source  \
```

```
0       2018-12-16 09:30:07  America/New_York  Haymarket Square
1       2018-11-27 02:00:23  America/New_York  Haymarket Square
2       2018-11-28 01:00:22  America/New_York  Haymarket Square
3       2018-11-30 04:53:02  America/New_York  Haymarket Square
4       2018-11-29 03:49:20  America/New_York  Haymarket Square
…                       …                 …                 …
693066  2018-12-01 23:53:05  America/New_York          West End
693067  2018-12-01 23:53:05  America/New_York          West End
693068  2018-12-01 23:53:05  America/New_York          West End
693069  2018-12-01 23:53:05  America/New_York          West End
693070  2018-12-01 23:53:05  America/New_York          West End

          destination cab_type  … precipIntensityMax uvIndexTime  \
0       North Station     Lyft  …              0.1276  1544979600
1       North Station     Lyft  …              0.1300  1543251600
2       North Station     Lyft  …              0.1064  1543338000
3       North Station     Lyft  …              0.0000  1543507200
4       North Station     Lyft  …              0.0001  1543420800
…                 …      …  …                   …          …
693066      North End     Uber  …              0.0000  1543683600
693067      North End     Uber  …              0.0000  1543683600
693068      North End     Uber  …              0.0000  1543683600
693069      North End     Uber  …              0.0000  1543683600
693070      North End     Uber  …              0.0000  1543683600

        temperatureMin  temperatureMinTime  temperatureMax  \
0                39.89          1545012000           43.68
1                40.49          1543233600           47.30
2                35.36          1543377600           47.55
3                34.67          1543550400           45.03
4                33.10          1543402800           42.18
…                …                 …              …
693066           31.42          1543658400           44.76
693067           31.42          1543658400           44.76
693068           31.42          1543658400           44.76
693069           31.42          1543658400           44.76
693070           31.42          1543658400           44.76

        temperatureMaxTime  apparentTemperatureMin  \
0               1544968800                   33.73
1               1543251600                   36.20
2               1543320000                   31.04
3               1543510800                   30.30
4               1543420800                   29.11
…                       …                   …
693066          1543690800                   27.77
693067          1543690800                   27.77
```

```
693068          1543690800                   27.77
693069          1543690800                   27.77
693070          1543690800                   27.77

        apparentTemperatureMinTime  apparentTemperatureMax  \
0                       1545012000                   38.07
1                       1543291200                   43.92
2                       1543377600                   44.12
3                       1543550400                   38.53
4                       1543392000                   35.75
…                              …                       …
693066                  1543658400                   44.09
693067                  1543658400                   44.09
693068                  1543658400                   44.09
693069                  1543658400                   44.09
693070                  1543658400                   44.09

       apparentTemperatureMaxTime
0                       1544958000
1                       1543251600
2                       1543320000
3                       1543510800
4                       1543420800
…                              …
693066                  1543690800
693067                  1543690800
693068                  1543690800
693069                  1543690800
693070                  1543690800

[693071 rows x 57 columns]>
```

[14]:
```
uber.isnull().sum()
```

[14]:
```
id                              0
timestamp                       0
hour                            0
day                             0
month                           0
datetime                        0
timezone                        0
source                          0
destination                     0
cab_type                        0
product_id                      0
name                            0
price                       55095
```

```
distance                        0
surge_multiplier                0
latitude                        0
longitude                       0
temperature                     0
apparentTemperature             0
short_summary                   0
long_summary                    0
precipIntensity                 0
precipProbability               0
humidity                        0
windSpeed                       0
windGust                        0
windGustTime                    0
visibility                      0
temperatureHigh                 0
temperatureHighTime             0
temperatureLow                  0
temperatureLowTime              0
apparentTemperatureHigh         0
apparentTemperatureHighTime     0
apparentTemperatureLow          0
apparentTemperatureLowTime      0
icon                            0
dewPoint                        0
pressure                        0
windBearing                     0
cloudCover                      0
uvIndex                         0
visibility.1                    0
ozone                           0
sunriseTime                     0
sunsetTime                      0
moonPhase                       0
precipIntensityMax              0
uvIndexTime                     0
temperatureMin                  0
temperatureMinTime              0
temperatureMax                  0
temperatureMaxTime              0
apparentTemperatureMin          0
apparentTemperatureMinTime      0
apparentTemperatureMax          0
apparentTemperatureMaxTime      0
dtype: int64
```

2 describe features

[ ]:

[114]:
```
uber.dtypes
```

[114]:
```
id                             object
timestamp                     float64
hour                            int64
day                             int64
month                           int64
datetime                       object
timezone                       object
source                         object
destination                    object
cab_type                       object
product_id                     object
name                           object
price                         float64
distance                      float64
surge_multiplier              float64
latitude                      float64
longitude                     float64
temperature                   float64
apparentTemperature           float64
short_summary                  object
long_summary                   object
precipIntensity               float64
precipProbability             float64
humidity                      float64
windSpeed                     float64
windGust                      float64
windGustTime                    int64
visibility                    float64
temperatureHigh               float64
temperatureHighTime             int64
temperatureLow                float64
temperatureLowTime              int64
apparentTemperatureHigh       float64
apparentTemperatureHighTime     int64
apparentTemperatureLow        float64
apparentTemperatureLowTime      int64
icon                           object
dewPoint                      float64
pressure                      float64
windBearing                     int64
cloudCover                    float64
```

```
uvIndex                          int64
visibility.1                   float64
ozone                          float64
sunriseTime                      int64
sunsetTime                       int64
moonPhase                      float64
precipIntensityMax             float64
uvIndexTime                      int64
temperatureMin                 float64
temperatureMinTime               int64
temperatureMax                 float64
temperatureMaxTime               int64
apparentTemperatureMin         float64
apparentTemperatureMinTime       int64
apparentTemperatureMax         float64
apparentTemperatureMaxTime       int64
dtype: object
```

[115]:
```
uber.describe()
```

[115]:

	timestamp	hour	day	month
count	6.930710e+05	693071.000000	693071.000000	693071.000000
mean	1.544046e+09	11.619137	17.794365	11.586684
std	6.891925e+05	6.948114	9.982286	0.492429
min	1.543204e+09	0.000000	1.000000	11.000000
25%	1.543444e+09	6.000000	13.000000	11.000000
50%	1.543737e+09	12.000000	17.000000	12.000000
75%	1.544828e+09	18.000000	28.000000	12.000000
max	1.545161e+09	23.000000	30.000000	12.000000

	price	distance	surge_multiplier	latitude
count	637976.000000	693071.000000	693071.000000	693071.000000
mean	16.545125	2.189430	1.013870	42.338172
std	9.324359	1.138937	0.091641	0.047840
min	2.500000	0.020000	1.000000	42.214800
25%	9.000000	1.280000	1.000000	42.350300
50%	13.500000	2.160000	1.000000	42.351900
75%	22.500000	2.920000	1.000000	42.364700
max	97.500000	7.860000	3.000000	42.366100

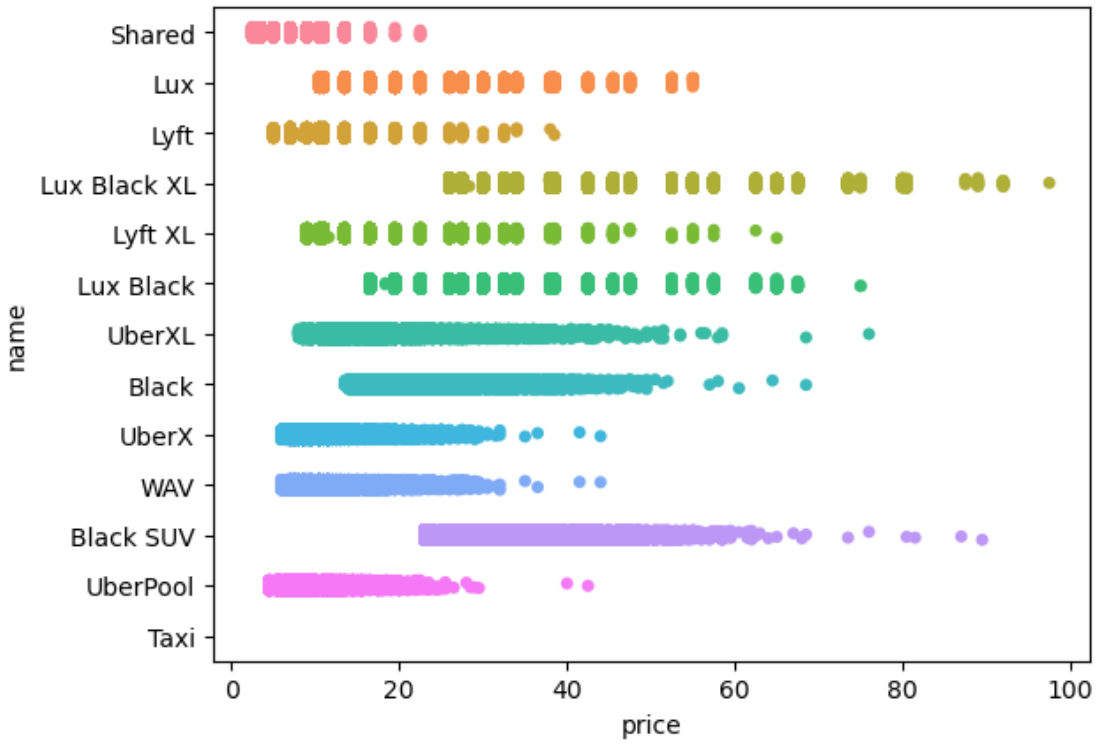
	longitude	temperature	…	precipIntensityMax	uvIndexTime
count	693071.000000	693071.000000	…	693071.000000	6.930710e+05
mean	-71.066151	39.584388	…	0.037374	1.544044e+09
std	0.020302	6.726084	…	0.055214	6.912028e+05
min	-71.105400	18.910000	…	0.000000	1.543162e+09
25%	-71.081000	36.450000	…	0.000000	1.543421e+09
50%	-71.063100	40.490000	…	0.000400	1.543770e+09

```
75%       -71.054200      43.580000  …            0.091600  1.544807e+09
max       -71.033000      57.220000  …            0.145900  1.545152e+09

       temperatureMin  temperatureMinTime  temperatureMax  temperatureMaxTime  \
count   693071.000000        6.930710e+05   693071.000000        6.930710e+05
mean        33.457774        1.544042e+09       45.261313        1.544047e+09
std          6.467224        6.901954e+05        5.645046        6.901353e+05
min         15.630000        1.543122e+09       33.510000        1.543154e+09
25%         30.170000        1.543399e+09       42.570000        1.543439e+09
50%         34.240000        1.543727e+09       44.680000        1.543788e+09
75%         38.880000        1.544789e+09       46.910000        1.544814e+09
max         43.100000        1.545192e+09       57.870000        1.545109e+09

       apparentTemperatureMin  apparentTemperatureMinTime  \
count           693071.000000                6.930710e+05
mean                29.731002                1.544048e+09
std                  7.110494                6.871862e+05
min                 11.810000                1.543136e+09
25%                 27.760000                1.543399e+09
50%                 30.130000                1.543745e+09
75%                 35.710000                1.544789e+09
max                 40.050000                1.545134e+09

       apparentTemperatureMax  apparentTemperatureMaxTime
count           693071.000000                6.930710e+05
mean                41.997343                1.544048e+09
std                  6.936841                6.910777e+05
min                 28.950000                1.543187e+09
25%                 36.570000                1.543439e+09
50%                 40.950000                1.543788e+09
75%                 44.120000                1.544818e+09
max                 57.200000                1.545109e+09

[8 rows x 46 columns]
```

[ ]:

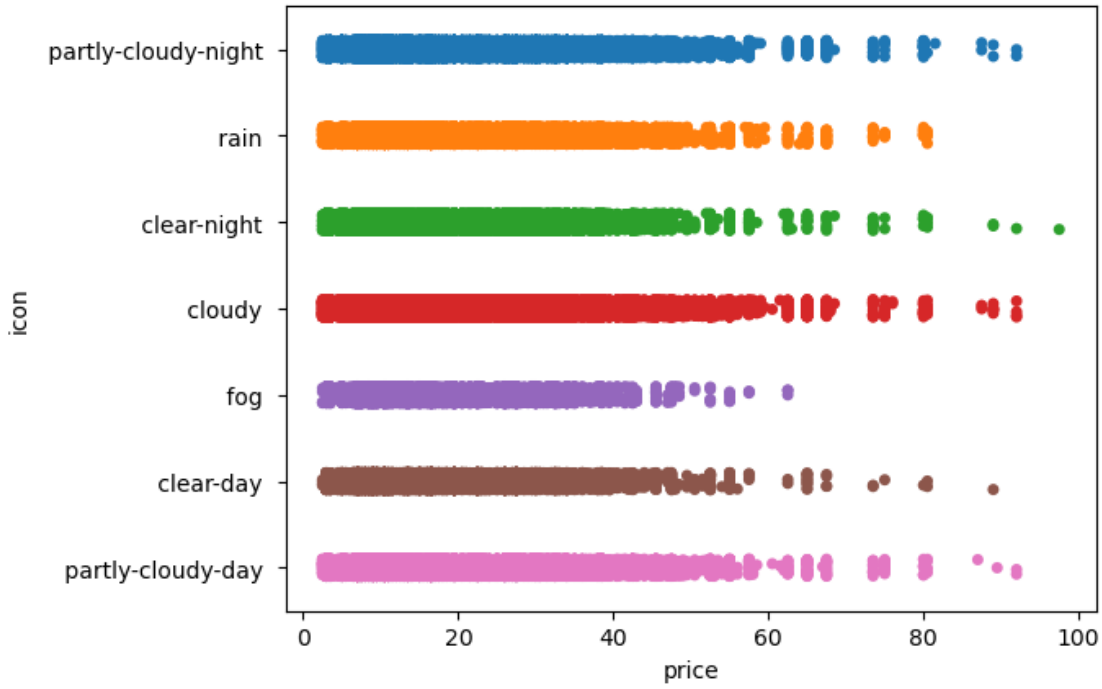
[116]:
```
sns.stripplot(data=uber, x='price', y='name')
```

[116]:
```
<AxesSubplot:xlabel='price', ylabel='name'>
```



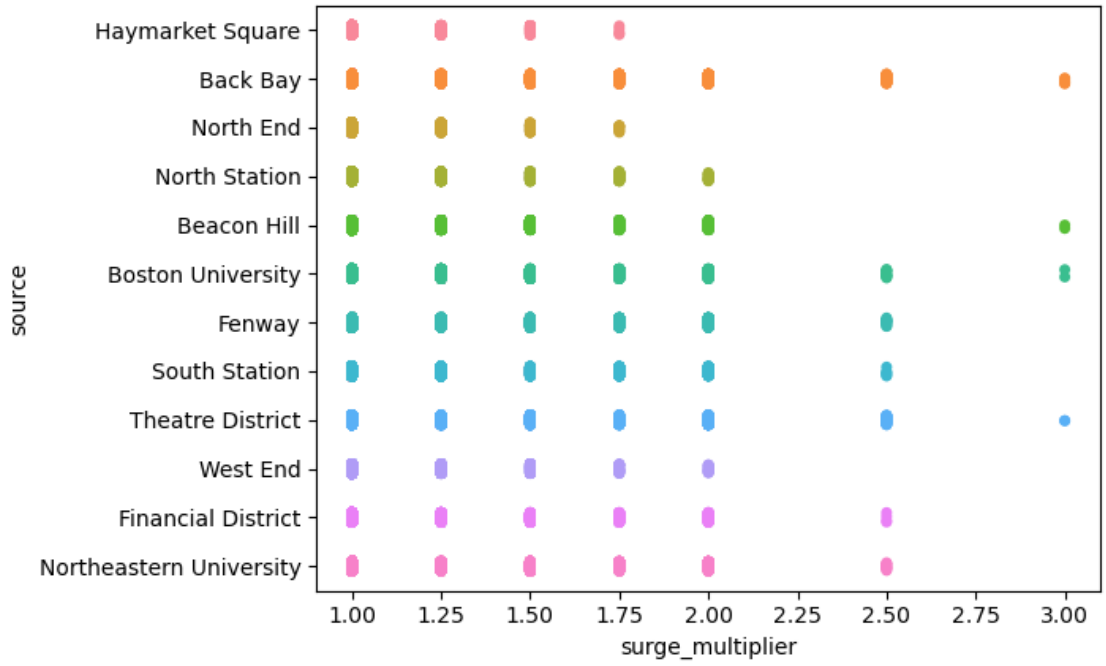
[117]:
```
sns.stripplot(data=uber, x='price', y='icon')
```

[117]:
```
<AxesSubplot:xlabel='price', ylabel='icon'>
```



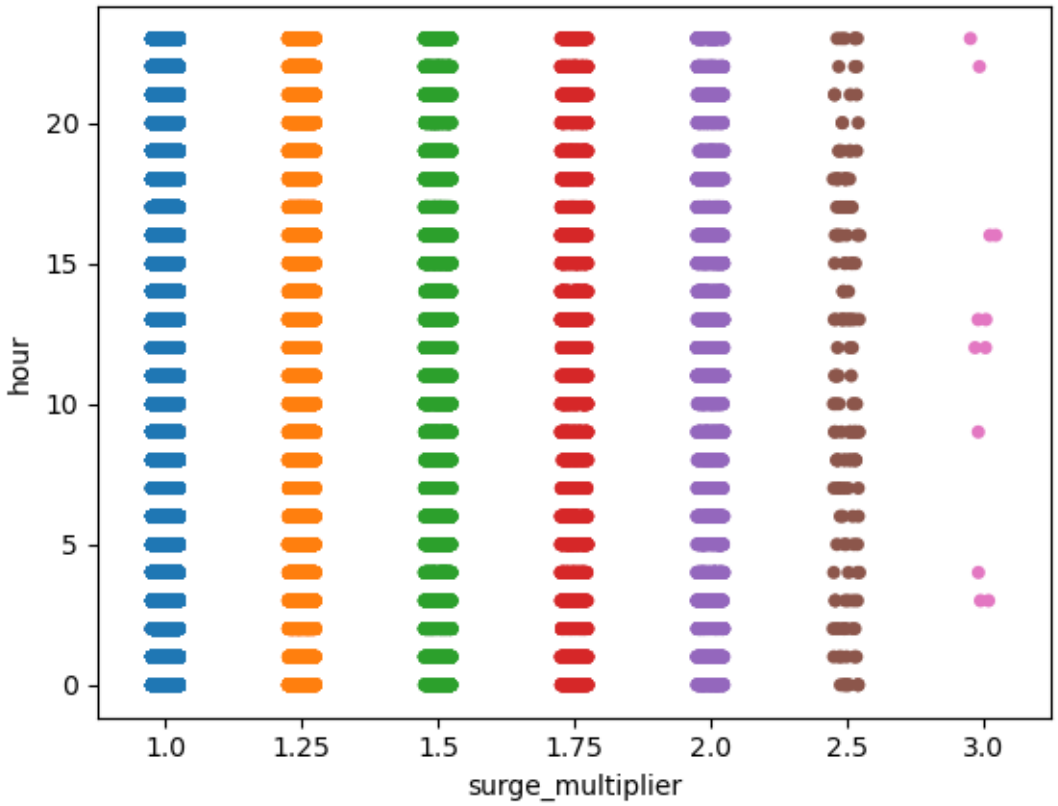
[121]:
```
sns.stripplot(data=uber, x='surge_multiplier', y='source')
sns
```

[121]:
```
<module 'seaborn' from 'C:\\ProgramData\\Anaconda3\\lib\\site-
packages\\seaborn\\__init__.py'>
```



[122]:
```
sns.stripplot(data=uber, x='surge_multiplier', y='hour')
```

[122]:
```
<AxesSubplot:xlabel='surge_multiplier', ylabel='hour'>
```



[22]:
```
#Converting Timestamp to Datetime value
```

[123]:
```
uber['timestamp'] = uber['timestamp'].astype('int')
```

[20]:
```
uber['timestamp'].head()
```

[20]:
```
0    1544952607
1    1543284023
2    1543366822
3    1543553582
4    1543463360
Name: timestamp, dtype: int32
```

[124]:
```
from datetime import datetime
timestamp1 = 1544952607
timestamp2 = 1543284023
timestamp3 = 1543366822
timestamp4 = 1543553582
timestamp5 = 1543463360
dt_object1 = datetime.fromtimestamp(timestamp1)
dt_object2 = datetime.fromtimestamp(timestamp2)
dt_object3 = datetime.fromtimestamp(timestamp3)
dt_object4 = datetime.fromtimestamp(timestamp4)
dt_object5 = datetime.fromtimestamp(timestamp5)

print("dt_object =", dt_object1)
print("dt_object =", dt_object2)
print("dt_object =", dt_object3)
print("dt_object =", dt_object4)
print("dt_object =", dt_object5)
```

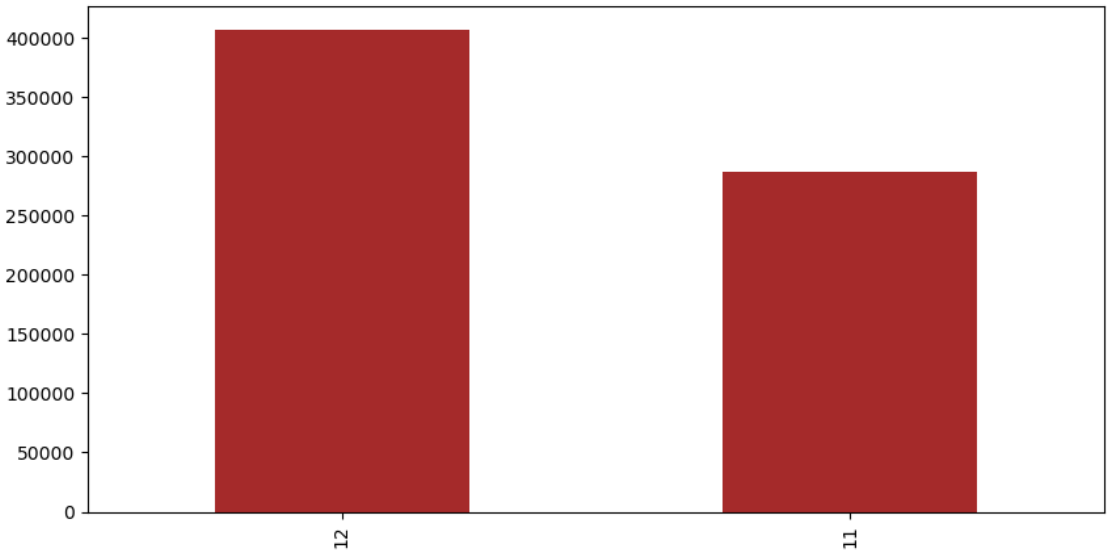
```
dt_object = 2018-12-16 15:00:07
dt_object = 2018-11-27 07:30:23
dt_object = 2018-11-28 06:30:22
dt_object = 2018-11-30 10:23:02
dt_object = 2018-11-29 09:19:20
```

#So by this timestamp to datetime conversion we get to know that, our data is of the year 2018 and in the month of november and december only

3 Bar plots

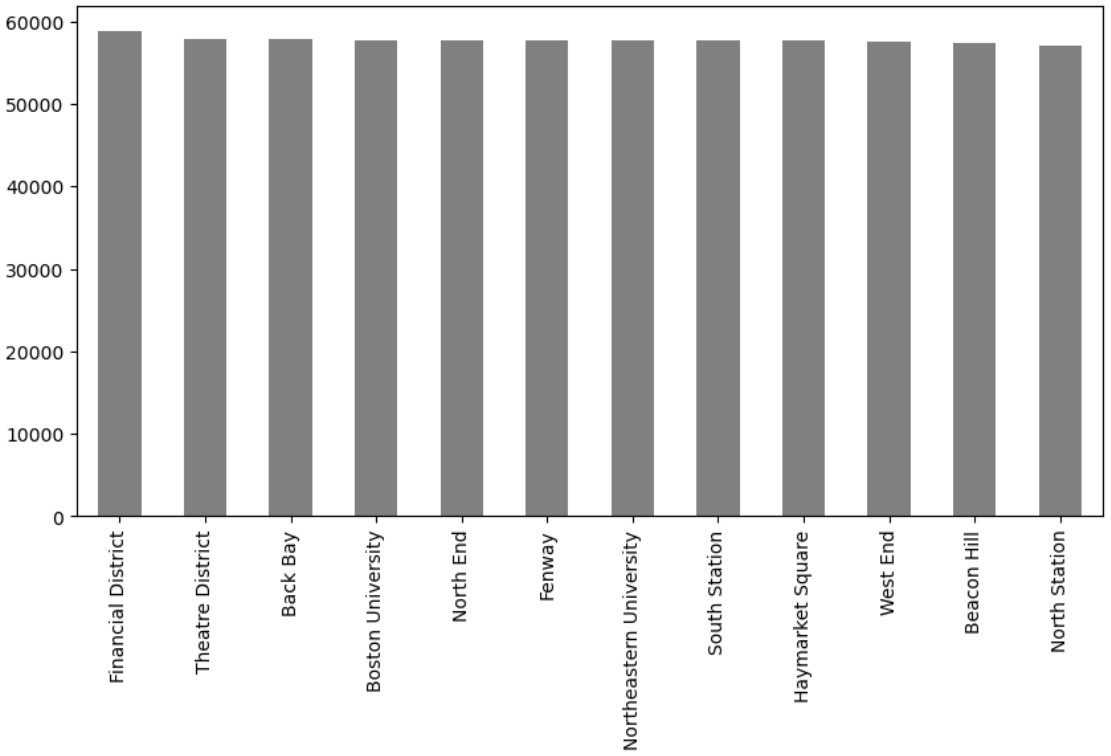
[125]:
```
uber['month'].value_counts().plot(kind='bar', figsize=(10,5), color='brown')
```

[125]:
```
<AxesSubplot:>
```



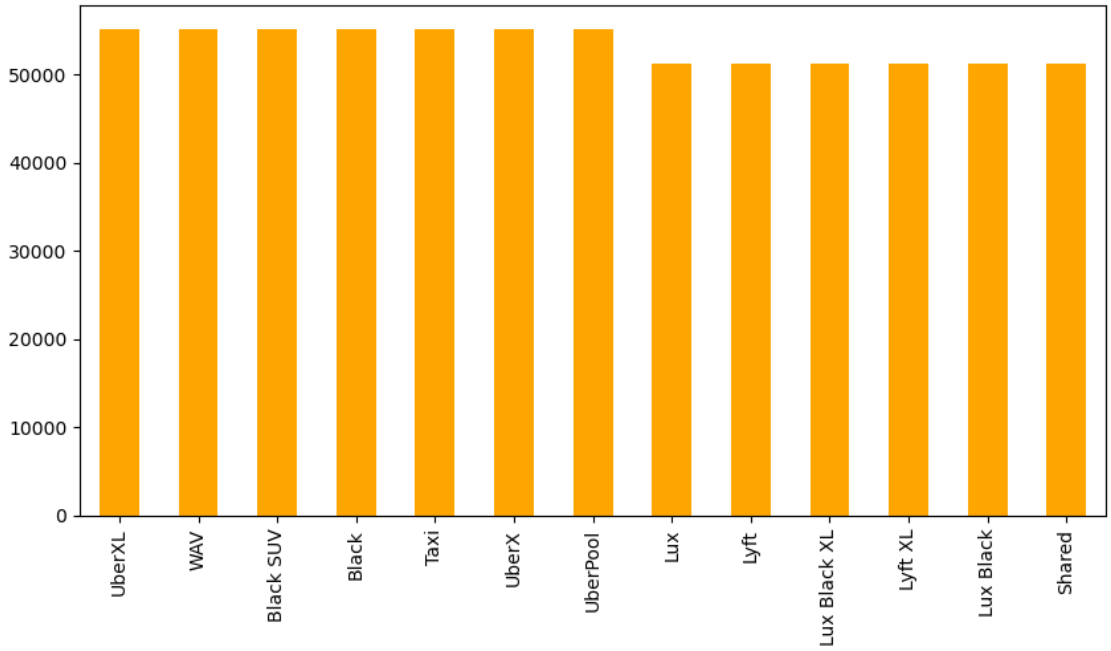
[126]:
```
uber['source'].value_counts().plot(kind='bar', figsize=(10,5), color='grey')
```

[126]: <AxesSubplot:>



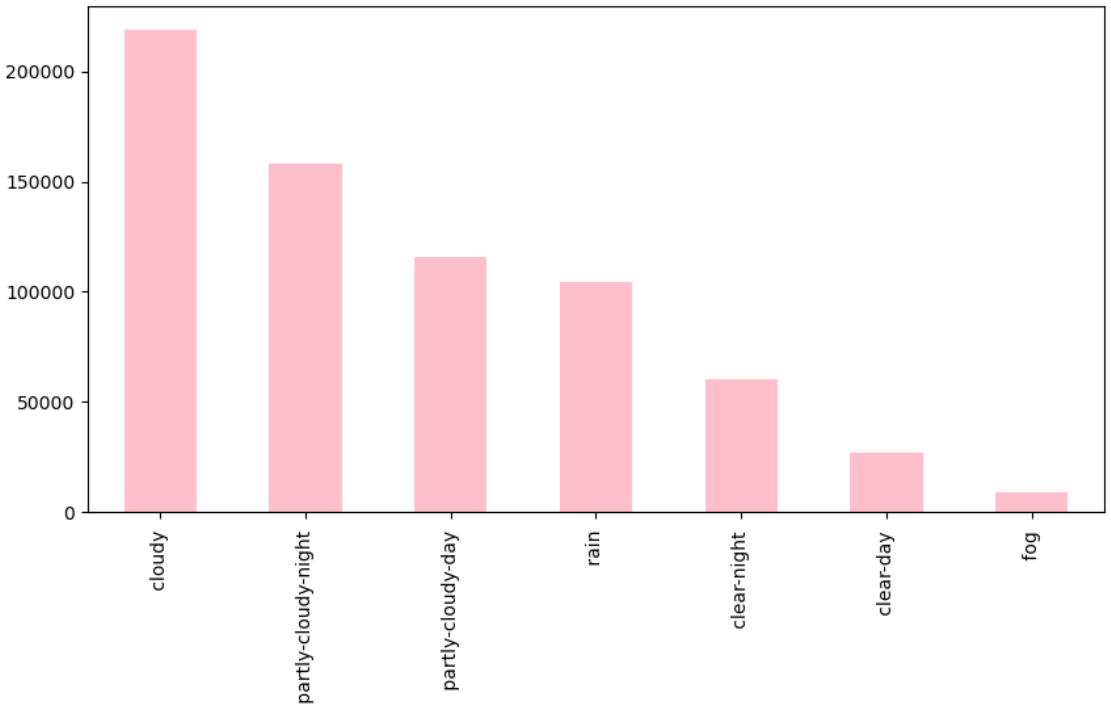
[127]:
```
uber['name'].value_counts().plot(kind='bar', figsize=(10,5), color='orange')
```

[127]: <AxesSubplot:>



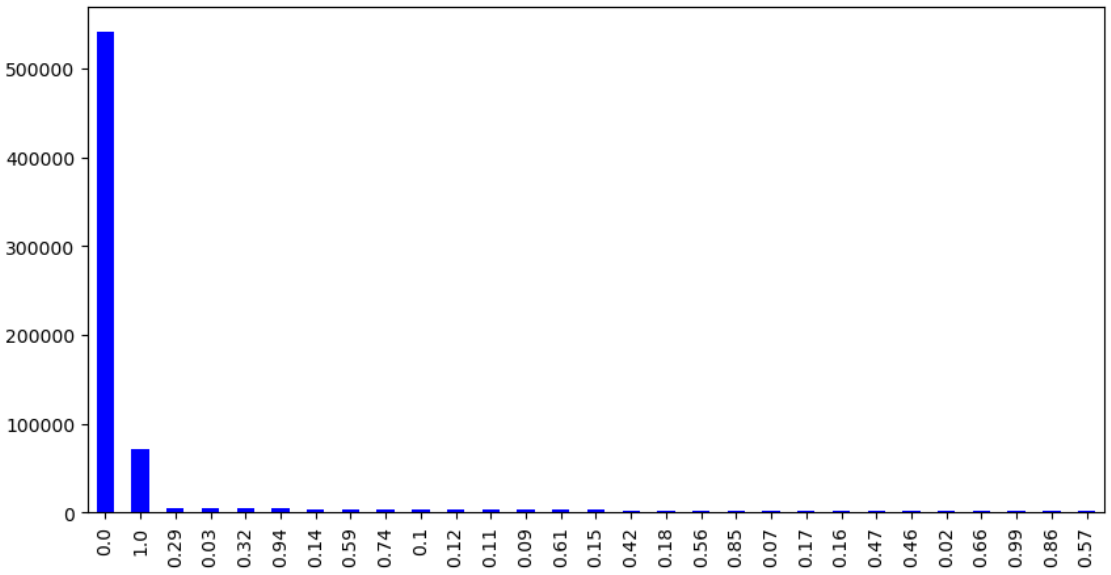
[128]:
```
uber['icon'].value_counts().plot(kind='bar', figsize=(10,5), color='pink')
```

[128]: <AxesSubplot:>



[129]:
```
uber['precipProbability'].value_counts().plot(kind='bar', figsize=(10,5),␣
↪color='blue')
```

[129]: <AxesSubplot:>



4 Label Encoding

[130]:
```
# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()
```

[131]:
```
uber.dtypes
```

[131]:
```
id                              object
timestamp                        int32
hour                             int64
day                              int64
month                            int64
datetime                        object
timezone                        object
source                          object
destination                     object
cab_type                        object
product_id                      object
name                            object
price                          float64
distance                       float64
surge_multiplier               float64
latitude                       float64
longitude                      float64
temperature                    float64
apparentTemperature            float64
short_summary                   object
long_summary                    object
precipIntensity                float64
precipProbability              float64
humidity                       float64
windSpeed                      float64
windGust                       float64
windGustTime                     int64
visibility                     float64
temperatureHigh                float64
temperatureHighTime              int64
temperatureLow                 float64
temperatureLowTime               int64
apparentTemperatureHigh        float64
apparentTemperatureHighTime      int64
apparentTemperatureLow         float64
apparentTemperatureLowTime       int64
icon                            object
dewPoint                       float64
pressure                       float64
windBearing                      int64
```

```
cloudCover                    float64
uvIndex                         int64
visibility.1                  float64
ozone                         float64
sunriseTime                     int64
sunsetTime                      int64
moonPhase                     float64
precipIntensityMax            float64
uvIndexTime                     int64
temperatureMin                float64
temperatureMinTime              int64
temperatureMax                float64
temperatureMaxTime              int64
apparentTemperatureMin        float64
apparentTemperatureMinTime      int64
apparentTemperatureMax        float64
apparentTemperatureMaxTime      int64
dtype: object
```

[132]:
```
uber['id']= label_encoder.fit_transform(uber['id'])
uber['datetime']= label_encoder.fit_transform(uber['datetime'])
uber['timezone']= label_encoder.fit_transform(uber['timezone'])
uber['destination']= label_encoder.fit_transform(uber['destination'])
uber['product_id']= label_encoder.fit_transform(uber['product_id'])
uber['short_summary']= label_encoder.fit_transform(uber['short_summary'])
uber['long_summary']= label_encoder.fit_transform(uber['long_summary'])
```

[133]:
```
uber['name']= label_encoder.fit_transform(uber['name'])

print("Class mapping of Name: ")
for i, item in enumerate(label_encoder.classes_):
    print(item, "-->", i)
```

```
Class mapping of Name:
Black --> 0
Black SUV --> 1
Lux --> 2
Lux Black --> 3
Lux Black XL --> 4
Lyft --> 5
Lyft XL --> 6
Shared --> 7
Taxi --> 8
UberPool --> 9
UberX --> 10
UberXL --> 11
WAV --> 12
```

[134]:
```
uber['source']= label_encoder.fit_transform(uber['source'])

print("Class mapping of Source: ")
for i, item in enumerate(label_encoder.classes_):
    print(item, "-->", i)
```

```
Class mapping of Source:
Back Bay --> 0
Beacon Hill --> 1
Boston University --> 2
Fenway --> 3
Financial District --> 4
Haymarket Square --> 5
North End --> 6
North Station --> 7
Northeastern University --> 8
South Station --> 9
Theatre District --> 10
West End --> 11
```

[135]:
```
uber['icon']= label_encoder.fit_transform(uber['icon'])

print("Class mapping of Icon: ")
for i, item in enumerate(label_encoder.classes_):
    print(item, "-->", i)
```

```
Class mapping of Icon:
 clear-day  --> 0
 clear-night  --> 1
 cloudy  --> 2
 fog  --> 3
 partly-cloudy-day  --> 4
 partly-cloudy-night  --> 5
 rain  --> 6
```

[136]:
```
uber.dtypes
```

[136]:
```
id                          int32
timestamp                   int32
hour                        int64
day                         int64
month                       int64
datetime                    int32
timezone                    int32
source                      int32
destination                 int32
cab_type                   object
product_id                  int32
```

```
name                          int32
price                       float64
distance                    float64
surge_multiplier            float64
latitude                    float64
longitude                   float64
temperature                 float64
apparentTemperature         float64
short_summary                 int32
long_summary                  int32
precipIntensity             float64
precipProbability           float64
humidity                    float64
windSpeed                   float64
windGust                    float64
windGustTime                  int64
visibility                  float64
temperatureHigh             float64
temperatureHighTime           int64
temperatureLow              float64
temperatureLowTime            int64
apparentTemperatureHigh     float64
apparentTemperatureHighTime   int64
apparentTemperatureLow      float64
apparentTemperatureLowTime    int64
icon                          int32
dewPoint                    float64
pressure                    float64
windBearing                   int64
cloudCover                  float64
uvIndex                       int64
visibility.1                float64
ozone                       float64
sunriseTime                   int64
sunsetTime                    int64
moonPhase                   float64
precipIntensityMax          float64
uvIndexTime                   int64
temperatureMin              float64
temperatureMinTime            int64
temperatureMax              float64
temperatureMaxTime            int64
apparentTemperatureMin      float64
apparentTemperatureMinTime    int64
apparentTemperatureMax      float64
apparentTemperatureMaxTime    int64
dtype: object
```

[137]:
```
uber.head()
```

[137]:
```
        id   timestamp  hour  day  month  datetime  timezone  source  \
0  179271  1544952607     9   16     12     25351         0       5
1  205021  1543284023     2   27     11       961         0       5
2  411506  1543366822     1   28     11      2534         0       5
3  527263  1543553582     4   30     11      6988         0       5
4  606526  1543463360     3   29     11      4400         0       5

   destination cab_type  …  precipIntensityMax  uvIndexTime  temperatureMin  \
0            7     Lyft  …              0.1276   1544979600           39.89
1            7     Lyft  …              0.1300   1543251600           40.49
2            7     Lyft  …              0.1064   1543338000           35.36
3            7     Lyft  …              0.0000   1543507200           34.67
4            7     Lyft  …              0.0001   1543420800           33.10

   temperatureMinTime  temperatureMax  temperatureMaxTime  \
0          1545012000           43.68          1544968800
1          1543233600           47.30          1543251600
2          1543377600           47.55          1543320000
3          1543550400           45.03          1543510800
4          1543402800           42.18          1543420800

   apparentTemperatureMin  apparentTemperatureMinTime  apparentTemperatureMax  \
0                   33.73                  1545012000                   38.07
1                   36.20                  1543291200                   43.92
2                   31.04                  1543377600                   44.12
3                   30.30                  1543550400                   38.53
4                   29.11                  1543392000                   35.75

   apparentTemperatureMaxTime
0                  1544958000
1                  1543251600
2                  1543320000
3                  1543510800
4                  1543420800

[5 rows x 57 columns]
```

5 Filling NAN Values

[138]:
```
uber.isnull().sum()
```

[138]:
```
id                          0
timestamp                   0
hour                        0
```

day	0
month	0
datetime	0
timezone	0
source	0
destination	0
cab_type	0
product_id	0
name	0
price	55095
distance	0
surge_multiplier	0
latitude	0
longitude	0
temperature	0
apparentTemperature	0
short_summary	0
long_summary	0
precipIntensity	0
precipProbability	0
humidity	0
windSpeed	0
windGust	0
windGustTime	0
visibility	0
temperatureHigh	0
temperatureHighTime	0
temperatureLow	0
temperatureLowTime	0
apparentTemperatureHigh	0
apparentTemperatureHighTime	0
apparentTemperatureLow	0
apparentTemperatureLowTime	0
icon	0
dewPoint	0
pressure	0
windBearing	0
cloudCover	0
uvIndex	0
visibility.1	0
ozone	0
sunriseTime	0
sunsetTime	0
moonPhase	0
precipIntensityMax	0
uvIndexTime	0
temperatureMin	0

```
temperatureMinTime                0
temperatureMax                    0
temperatureMaxTime                0
apparentTemperatureMin            0
apparentTemperatureMinTime        0
apparentTemperatureMax            0
apparentTemperatureMaxTime        0
dtype: int64
```

[139]:
```
uber['price'].median()
```

[139]:
```
13.5
```

[140]:
```
uber["price"].fillna(10.5, inplace = True)
```

[141]:
```
uber.isnull().sum()
```

[141]:
```
id                          0
timestamp                   0
hour                        0
day                         0
month                       0
datetime                    0
timezone                    0
source                      0
destination                 0
cab_type                    0
product_id                  0
name                        0
price                       0
distance                    0
surge_multiplier            0
latitude                    0
longitude                   0
temperature                 0
apparentTemperature         0
short_summary               0
long_summary                0
precipIntensity             0
precipProbability           0
humidity                    0
windSpeed                   0
windGust                    0
windGustTime                0
visibility                  0
temperatureHigh             0
temperatureHighTime         0
```

```
temperatureLow                0
temperatureLowTime            0
apparentTemperatureHigh       0
apparentTemperatureHighTime   0
apparentTemperatureLow        0
apparentTemperatureLowTime    0
icon                          0
dewPoint                      0
pressure                      0
windBearing                   0
cloudCover                    0
uvIndex                       0
visibility.1                  0
ozone                         0
sunriseTime                   0
sunsetTime                    0
moonPhase                     0
precipIntensityMax            0
uvIndexTime                   0
temperatureMin                0
temperatureMinTime            0
temperatureMax                0
temperatureMaxTime            0
apparentTemperatureMin        0
apparentTemperatureMinTime    0
apparentTemperatureMax        0
apparentTemperatureMaxTime    0
dtype: int64
```

[142]:
```
uber['price'].dtype
```

[142]:
```
dtype('float64')
```

[143]:
```
uber['price'] = uber['price'].astype(int)
```

[144]:
```
uber['price'].dtype
```

[144]:
```
dtype('int32')
```

[145]:
```
uber['price'].head()
```

[145]:
```
0     5
1    11
2     7
3    26
4     9
Name: price, dtype: int32
```

6 Recursive Feature Elimination

[146]:
```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.feature_selection import RFE
```

[147]:
```
X = uber.drop('price', axis = 1)
y = uber['price']
```

[148]:
```
X = uber.drop('cab_type', axis = 1)

y = uber.drop('cab_type', axis = 1)
```

[149]:
```
X.head()
```

[149]:

	id	timestamp	hour	day	month	datetime	timezone	source	\
0	179271	1544952607	9	16	12	25351	0	5	
1	205021	1543284023	2	27	11	961	0	5	
2	411506	1543366822	1	28	11	2534	0	5	
3	527263	1543553582	4	30	11	6988	0	5	
4	606526	1543463360	3	29	11	4400	0	5	

	destination	product_id	…	precipIntensityMax	uvIndexTime	\
0	7	8	…	0.1276	1544979600	
1	7	12	…	0.1300	1543251600	
2	7	7	…	0.1064	1543338000	
3	7	10	…	0.0000	1543507200	
4	7	11	…	0.0001	1543420800	

	temperatureMin	temperatureMinTime	temperatureMax	temperatureMaxTime	\
0	39.89	1545012000	43.68	1544968800	
1	40.49	1543233600	47.30	1543251600	
2	35.36	1543377600	47.55	1543320000	
3	34.67	1543550400	45.03	1543510800	
4	33.10	1543402800	42.18	1543420800	

	apparentTemperatureMin	apparentTemperatureMinTime	apparentTemperatureMax	\
0	33.73	1545012000	38.07	
1	36.20	1543291200	43.92	
2	31.04	1543377600	44.12	
3	30.30	1543550400	38.53	

```
4                   29.11                   1543392000                   35.75

   apparentTemperatureMaxTime
0                  1544958000
1                  1543251600
2                  1543320000
3                  1543510800
4                  1543420800

[5 rows x 56 columns]
```

[150]:
```
y.head()
```

[150]:
```
       id   timestamp  hour  day  month  datetime  timezone  source  \
0  179271  1544952607     9   16     12     25351         0       5
1  205021  1543284023     2   27     11       961         0       5
2  411506  1543366822     1   28     11      2534         0       5
3  527263  1543553582     4   30     11      6988         0       5
4  606526  1543463360     3   29     11      4400         0       5

   destination  product_id  …  precipIntensityMax  uvIndexTime  \
0            7           8  …              0.1276   1544979600
1            7          12  …              0.1300   1543251600
2            7           7  …              0.1064   1543338000
3            7          10  …              0.0000   1543507200
4            7          11  …              0.0001   1543420800

   temperatureMin  temperatureMinTime  temperatureMax  temperatureMaxTime  \
0           39.89          1545012000           43.68          1544968800
1           40.49          1543233600           47.30          1543251600
2           35.36          1543377600           47.55          1543320000
3           34.67          1543550400           45.03          1543510800
4           33.10          1543402800           42.18          1543420800

   apparentTemperatureMin  apparentTemperatureMinTime  apparentTemperatureMax  \
0                   33.73                  1545012000                   38.07
1                   36.20                  1543291200                   43.92
2                   31.04                  1543377600                   44.12
3                   30.30                  1543550400                   38.53
4                   29.11                  1543392000                   35.75

   apparentTemperatureMaxTime
0                  1544958000
1                  1543251600
2                  1543320000
3                  1543510800
4                  1543420800
```

```
[5 rows x 56 columns]
```

[151]:
```
X.shape
```

[151]:
```
(693071, 56)
```

[152]:
```
y.shape
```

[152]:
```
(693071, 56)
```

[153]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,␣
↪random_state = 42)
```

[154]:
```
X_train.shape
```

[154]:
```
(554456, 56)
```

[155]:
```
X_test.shape
```

[155]:
```
(138615, 56)
```

[156]:
```
y_train.shape
```

[156]:
```
(554456, 56)
```

[157]:
```
y_test.shape
```

[157]:
```
(138615, 56)
```

7 Creating model

[158]:
```
reg = LinearRegression()
#Fitting training data
reg = reg.fit(X_train, y_train)
```

[159]:
```
reg.score(X_train, y_train)
```

[159]:
```
1.0
```

[160]:
```
reg1 = LinearRegression()
#Fitting training data
reg1 = reg1.fit(X_train, y_train)
reg1.score(X_train, y_train)
```

[160]:
```
1.0
```

[61]:
```
rfe = RFE(reg, verbose=1)
rfe = rfe.fit(X, y)
```

```
Fitting estimator with 56 features.
Fitting estimator with 55 features.
Fitting estimator with 54 features.
Fitting estimator with 53 features.
Fitting estimator with 52 features.
Fitting estimator with 51 features.
Fitting estimator with 50 features.
Fitting estimator with 49 features.
Fitting estimator with 48 features.
Fitting estimator with 47 features.
Fitting estimator with 46 features.
Fitting estimator with 45 features.
Fitting estimator with 44 features.
Fitting estimator with 43 features.
Fitting estimator with 42 features.
Fitting estimator with 41 features.
Fitting estimator with 40 features.
Fitting estimator with 39 features.
Fitting estimator with 38 features.
Fitting estimator with 37 features.
Fitting estimator with 36 features.
Fitting estimator with 35 features.
Fitting estimator with 34 features.
Fitting estimator with 33 features.
Fitting estimator with 32 features.
Fitting estimator with 31 features.
Fitting estimator with 30 features.
Fitting estimator with 29 features.
```

[62]:
```
XX = X[X.columns[rfe.support_]]
```

[63]:
```
XX.head()
```

[63]:

	hour	day	surge_multiplier	latitude	longitude	temperature	\
0	9	16	1.0	42.2148	-71.033	42.34	
1	2	27	1.0	42.2148	-71.033	43.58	
2	1	28	1.0	42.2148	-71.033	38.33	
3	4	30	1.0	42.2148	-71.033	34.38	
4	3	29	1.0	42.2148	-71.033	37.44	

	apparentTemperature	short_summary	long_summary	precipIntensity	…	\
0	37.12	4	9	0.0000	…	
1	37.35	8	10	0.1299	…	
2	32.93	0	2	0.0000	…	
3	29.63	0	6	0.0000	…	

```
4                30.88              6              4           0.0000  …

   dewPoint  pressure  cloudCover  uvIndex  moonPhase  precipIntensityMax  \
0     32.70   1021.98        0.72        0       0.30              0.1276
1     41.83   1003.97        1.00        0       0.64              0.1300
2     31.10    992.28        0.03        0       0.68              0.1064
3     26.64   1013.73        0.00        0       0.75              0.0000
4     28.61    998.36        0.44        0       0.72              0.0001

   temperatureMin  temperatureMax  apparentTemperatureMin  \
0           39.89           43.68                   33.73
1           40.49           47.30                   36.20
2           35.36           47.55                   31.04
3           34.67           45.03                   30.30
4           33.10           42.18                   29.11

   apparentTemperatureMax
0                   38.07
1                   43.92
2                   44.12
3                   38.53
4                   35.75

[5 rows x 28 columns]
```

[64]:
```
X_train, X_test, y_train, y_test = train_test_split(XX, y, test_size = 0.3,␣
↪random_state = 10,)
```

[65]:
```
X_train.shape
```

[65]:
```
(485149, 28)
```

[66]:
```
#Creating model
reg1 = LinearRegression()
#Fitting training data
reg1 = reg1.fit(X_train, y_train)
```

[67]:
```
reg1.score(X_train, y_train)
```

[67]:
```
0.8443504818646126
```

[90]:
```
rfe = RFE(reg, verbose=1)
rfe = rfe.fit(X, y)
```

```
Fitting estimator with 56 features.
Fitting estimator with 55 features.
Fitting estimator with 54 features.
Fitting estimator with 53 features.
```

```
Fitting estimator with 52 features.
Fitting estimator with 51 features.
Fitting estimator with 50 features.
Fitting estimator with 49 features.
Fitting estimator with 48 features.
Fitting estimator with 47 features.
Fitting estimator with 46 features.
Fitting estimator with 45 features.
Fitting estimator with 44 features.
Fitting estimator with 43 features.
Fitting estimator with 42 features.
Fitting estimator with 41 features.
Fitting estimator with 40 features.
Fitting estimator with 39 features.
Fitting estimator with 38 features.
Fitting estimator with 37 features.
Fitting estimator with 36 features.
Fitting estimator with 35 features.
Fitting estimator with 34 features.
Fitting estimator with 33 features.
Fitting estimator with 32 features.
Fitting estimator with 31 features.
Fitting estimator with 30 features.
Fitting estimator with 29 features.
```

[68]:
```
XX.columns
```

[68]:
```
Index(['hour', 'day', 'surge_multiplier', 'latitude', 'longitude',
       'temperature', 'apparentTemperature', 'short_summary', 'long_summary',
       'precipIntensity', 'precipProbability', 'humidity', 'windSpeed',
       'temperatureHigh', 'temperatureLow', 'apparentTemperatureHigh',
       'apparentTemperatureLow', 'icon', 'dewPoint', 'pressure', 'cloudCover',
       'uvIndex', 'moonPhase', 'precipIntensityMax', 'temperatureMin',
       'temperatureMax', 'apparentTemperatureMin', 'apparentTemperatureMax'],
      dtype='object')
```

[70]:
```
XX.shape
```

[70]:
```
(693071, 28)
```

[71]:
```
XX.head()
```

[71]:

	hour	day	surge_multiplier	latitude	longitude	temperature	\
0	9	16	1.0	42.2148	-71.033	42.34	
1	2	27	1.0	42.2148	-71.033	43.58	
2	1	28	1.0	42.2148	-71.033	38.33	
3	4	30	1.0	42.2148	-71.033	34.38	
4	3	29	1.0	42.2148	-71.033	37.44	

```
   apparentTemperature  short_summary  long_summary  precipIntensity  …  \
0                37.12              4             9           0.0000  …
1                37.35              8            10           0.1299  …
2                32.93              0             2           0.0000  …
3                29.63              0             6           0.0000  …
4                30.88              6             4           0.0000  …

   dewPoint  pressure  cloudCover  uvIndex  moonPhase  precipIntensityMax  \
0     32.70   1021.98        0.72        0       0.30              0.1276
1     41.83   1003.97        1.00        0       0.64              0.1300
2     31.10    992.28        0.03        0       0.68              0.1064
3     26.64   1013.73        0.00        0       0.75              0.0000
4     28.61    998.36        0.44        0       0.72              0.0001

   temperatureMin  temperatureMax  apparentTemperatureMin  \
0           39.89           43.68                   33.73
1           40.49           47.30                   36.20
2           35.36           47.55                   31.04
3           34.67           45.03                   30.30
4           33.10           42.18                   29.11

   apparentTemperatureMax
0                   38.07
1                   43.92
2                   44.12
3                   38.53
4                   35.75

[5 rows x 28 columns]
```

[84]:
```
features_drop = ['latitude', 'longitude', 'apparentTemperature',
       'long_summary', 'precipIntensity', 'humidity', 'windSpeed',
       'temperatureHigh', 'apparentTemperatureHigh',␣
 ↪'dewPoint','precipIntensityMax',
       'temperatureMax', 'apparentTemperatureMax', 'cloudCover', 'moonPhase']
uber_new = XX.drop(features_drop, axis=1)
```

[85]:
```
uber_new.head()
```

[85]:
```
   hour  day  surge_multiplier  temperature  short_summary  precipProbability  \
0     9   16               1.0        42.34              4                0.0
1     2   27               1.0        43.58              8                1.0
2     1   28               1.0        38.33              0                0.0
3     4   30               1.0        34.38              0                0.0
4     3   29               1.0        37.44              6                0.0
```

```
   temperatureLow  apparentTemperatureLow  icon  pressure  uvIndex  \
0           34.19                   27.39     5   1021.98        0
1           42.10                   36.20     6   1003.97        0
2           33.10                   29.11     1    992.28        0
3           28.90                   26.20     1   1013.73        0
4           36.71                   30.29     5    998.36        0

   temperatureMin  apparentTemperatureMin
0           39.89                   33.73
1           40.49                   36.20
2           35.36                   31.04
3           34.67                   30.30
4           33.10                   29.11
```

[86]:
```
uber_new.shape
```

[86]:
```
(693071, 13)
```

[87]:
```
surge_multiplier_mapping = {1.: 0, 1.25: 1, 1.5: 2, 1.75: 3, 2.:4}
uber_new['surge_multiplier'] = uber_new['surge_multiplier'].
 ↪map(surge_multiplier_mapping)
```

[88]:
```
uber_new.head()
```

[88]:
```
   hour  day  surge_multiplier  temperature  short_summary  precipProbability  \
0     9   16               0.0        42.34              4                0.0
1     2   27               0.0        43.58              8                1.0
2     1   28               0.0        38.33              0                0.0
3     4   30               0.0        34.38              0                0.0
4     3   29               0.0        37.44              6                0.0

   temperatureLow  apparentTemperatureLow  icon  pressure  uvIndex  \
0           34.19                   27.39     5   1021.98        0
1           42.10                   36.20     6   1003.97        0
2           33.10                   29.11     1    992.28        0
3           28.90                   26.20     1   1013.73        0
4           36.71                   30.29     5    998.36        0

   temperatureMin  apparentTemperatureMin
0           39.89                   33.73
1           40.49                   36.20
2           35.36                   31.04
3           34.67                   30.30
4           33.10                   29.11
```

[161]:
```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
```

8 Linear regression

[163]:
```
linear = LinearRegression()
linear.fit(X_train, y_train)
linear.score(X_test, y_test)
```

[163]: 1.0

9 Decision Tree

[164]:
```
decision = DecisionTreeRegressor(random_state = 0)
decision.fit(X_train , y_train)
decision.score(X_test, y_test)
```

[164]: 0.7795201640010058

10 Random Forest

[ ]:
```
random = RandomForestRegressor(n_estimators = 100, random_state = 0)
random.fit(X_train , y_train)
random.score(X_test, y_test)
```

10.1 Observation

Linear Regression: The linear regression algorithm has been applied to the Uber data analysis, achieving an accuracy score of 1.0. This indicates that the linear regression model is able to perfectly fit the data and predict the target variable accurately. This high accuracy suggests that there is a strong linear relationship between the input features and the target variable in the dataset.

Decision Tree: The decision tree algorithm has been applied to the Uber data analysis, achieving an accuracy score of 0.7 This indicates that the decision tree model is able to capture around 77.95% of the patterns and relationships in the data, leading to reasonably accurate predictions. The accuracy score suggests that the decision tree algorithm performs well in analyzing the Uber dataset, but there is still room for improvement compared to the perfect fit of the linear regression algorithm.

10.1.1 conclusion

Data analysis plays a pivotal role in Uber's operations, providing valuable insights that drive strategic decision-making, improve user experiences, optimize driver efficiency, and expand into new markets. By harnessing the power of data, Uber continues to innovate and transform the transportation industry, ensuring that its services remain efficient, reliable, and convenient for millions of riders and drivers worldwide.