***UBER\_DATA ASSIGNMENT***

**BUSINESS OBJECTIVE**

You have an emergency to reach a destination point, but unfortunately no cabs are available or maybe cancelled by cab driver. Most of us faced this situation in our real life, while travelling to Airport or Railway station etc. This will surely effect the revenue and reputation of Uber. Main objective of this case study is to find the actual reason (or) cause of unavailability of the cars. I will try to identify the this problem uber is facing and recommend ways to improve the situation.

Data attributes

There are six attributes associated with each request made by a customer:

1) Request id : A unique identifies of the request

2) Time of request(request timestamp): The date and time at which customer made the trip request

3) Drop-off time (drop timestamp): The drop-off date and time, in case the trip was completed

4) Pickup-up point: The point from which the request was made

5) Driver id: The unique identification number of the driver

6) Status of the request: The final status of the trip, that can be either completed , cancelled by the driver or no cars available

# Data cleaning

1)Cleaning the data is very much important in vizualizing and analyzing the data.

2)Dates and formats should be in proper format

**#Importing the required libraries**

import numpy as np

import pandas as pd

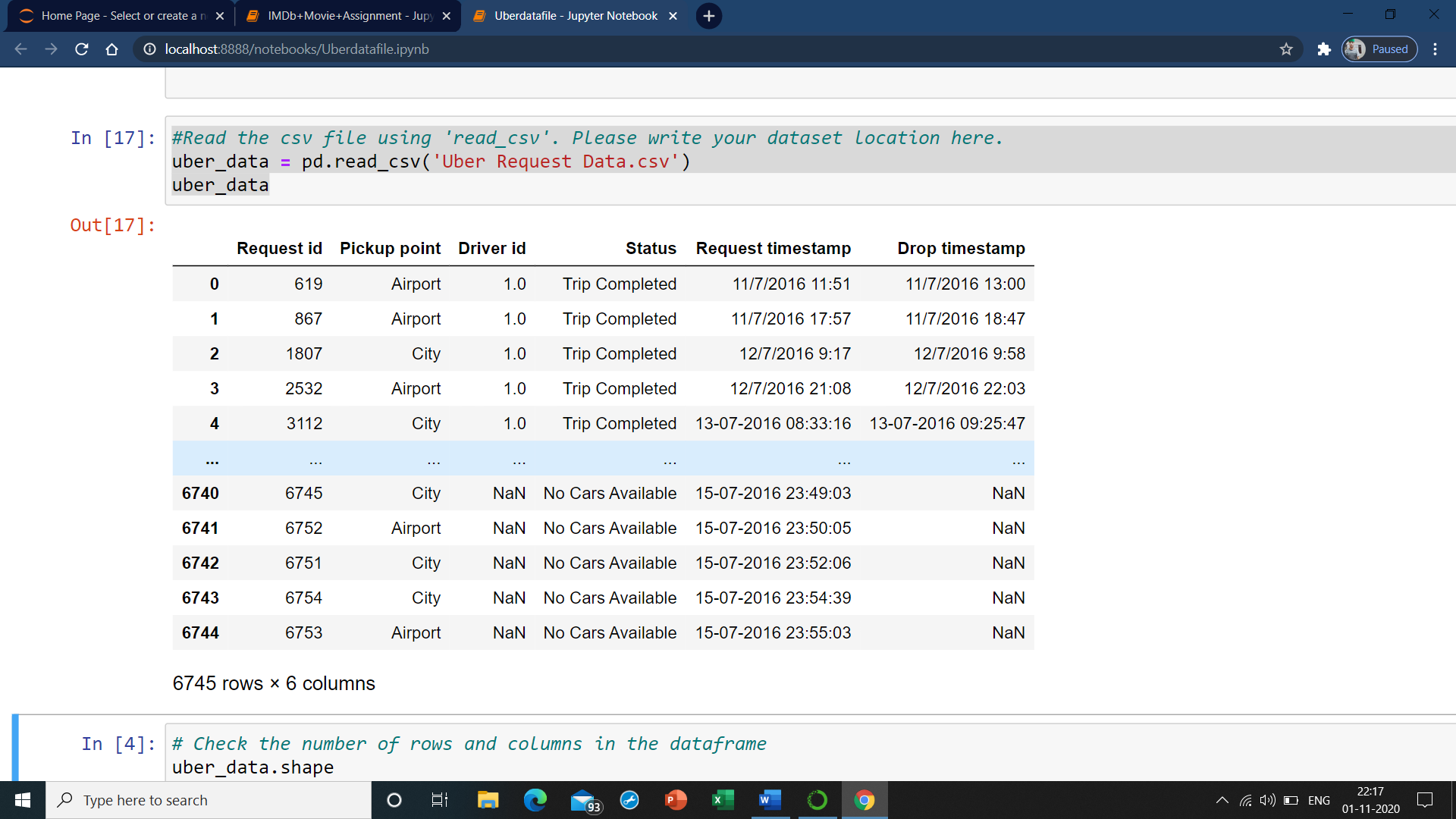
import matplotlib.pyplot as plt

import seaborn as sns

**#Read the csv file using 'read\_csv'. Please write your dataset location here**.

uber\_data = pd.read\_csv('Uber Request Data.csv')

uber\_data



**# Check the number of rows and columns in the dataframe**

uber\_data.shape

**Output:** **(6745, 6)**

**#index**

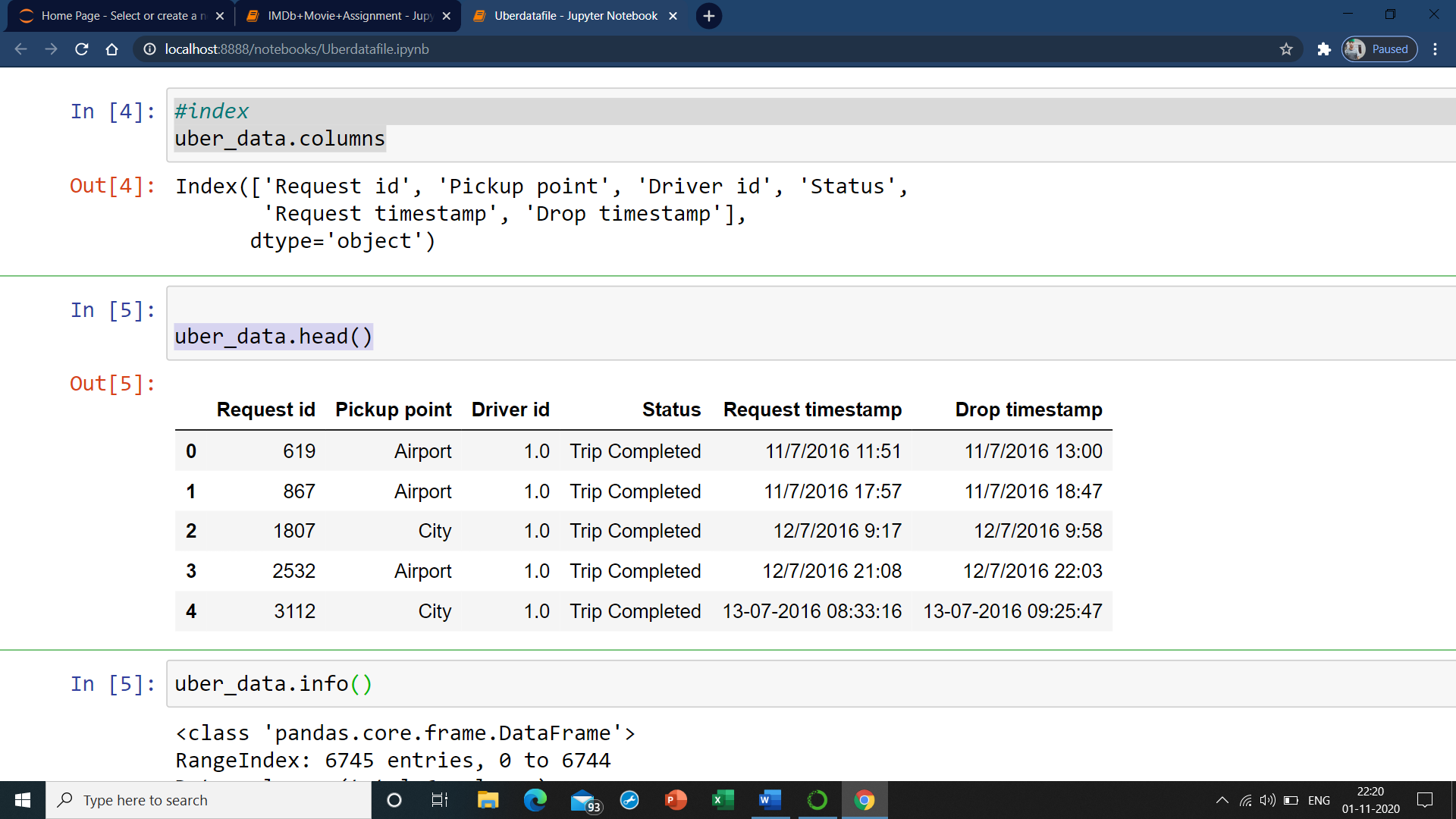
uber\_data.columns

**Output:** **Index(['Request id', 'Pickup point', 'Driver id', 'Status',**

**'Request timestamp', 'Drop timestamp'],**

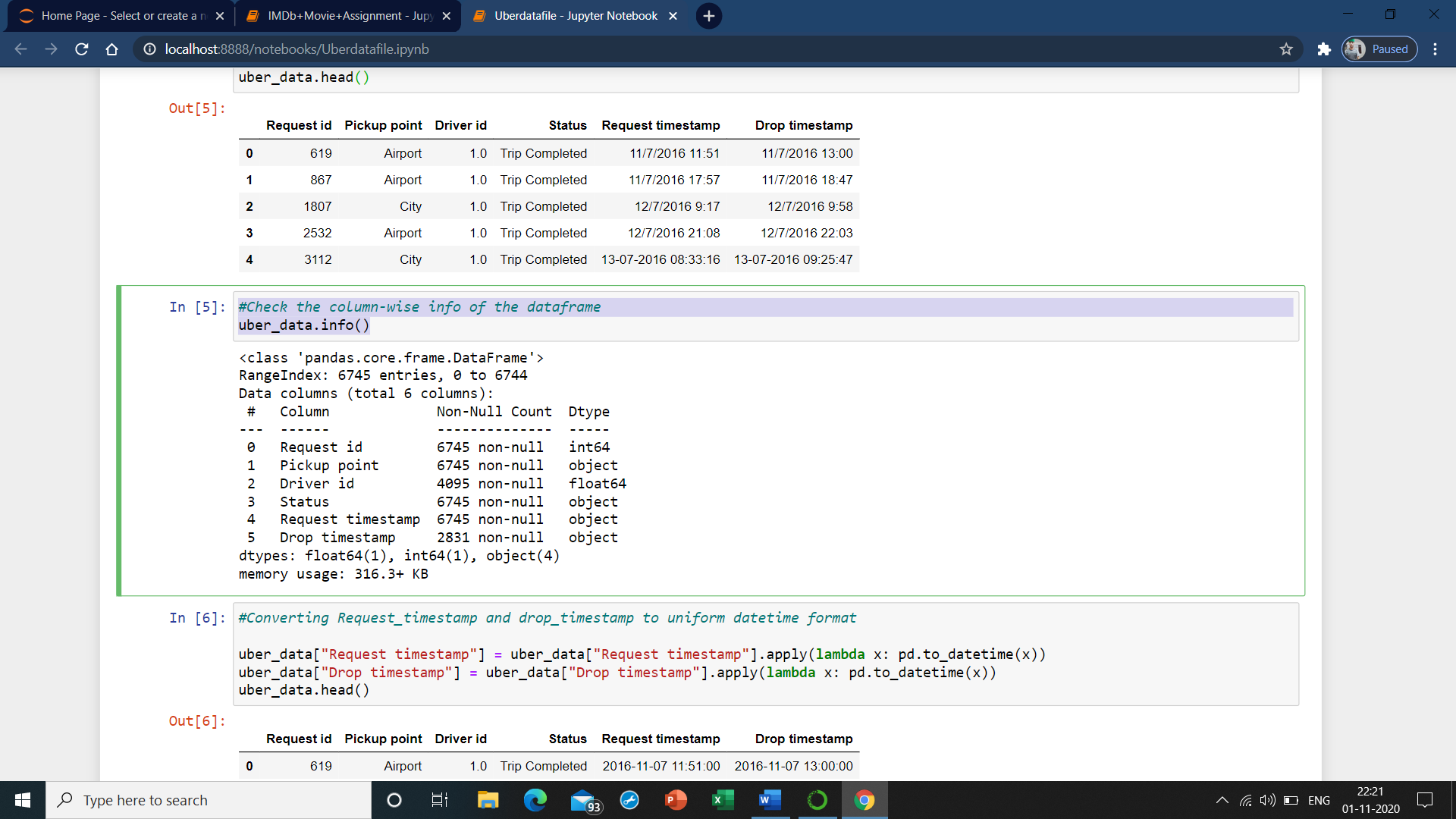
**dtype='object')**

**uber\_data.head()**



**#Check the column-wise info of the dataframe**

**uber\_data.info()**

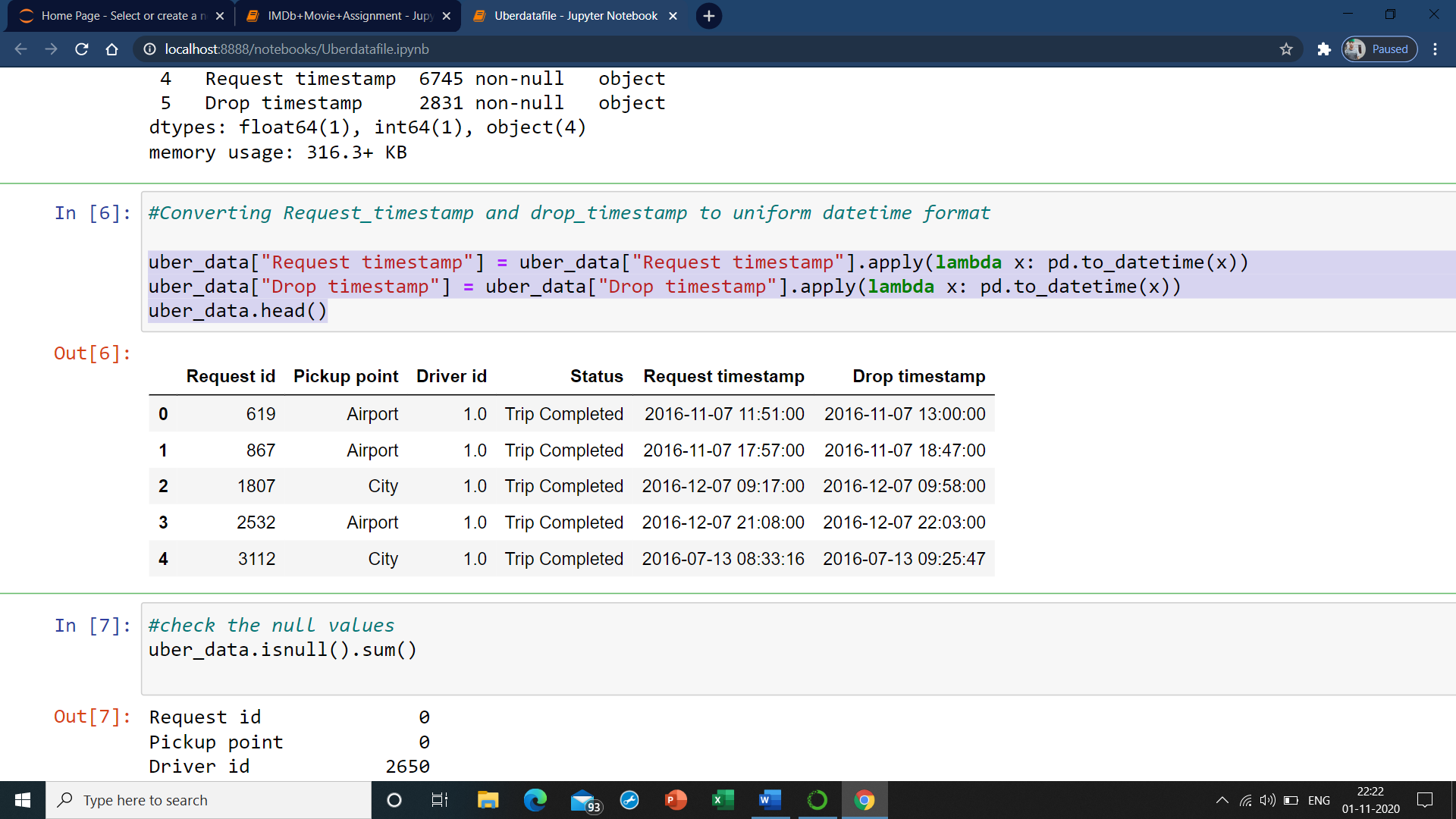


**#Converting Request\_timestamp and drop\_timestamp to uniform datetime format**

uber\_data["Request timestamp"] = uber\_data["Request timestamp"].apply(lambda x: pd.to\_datetime(x))

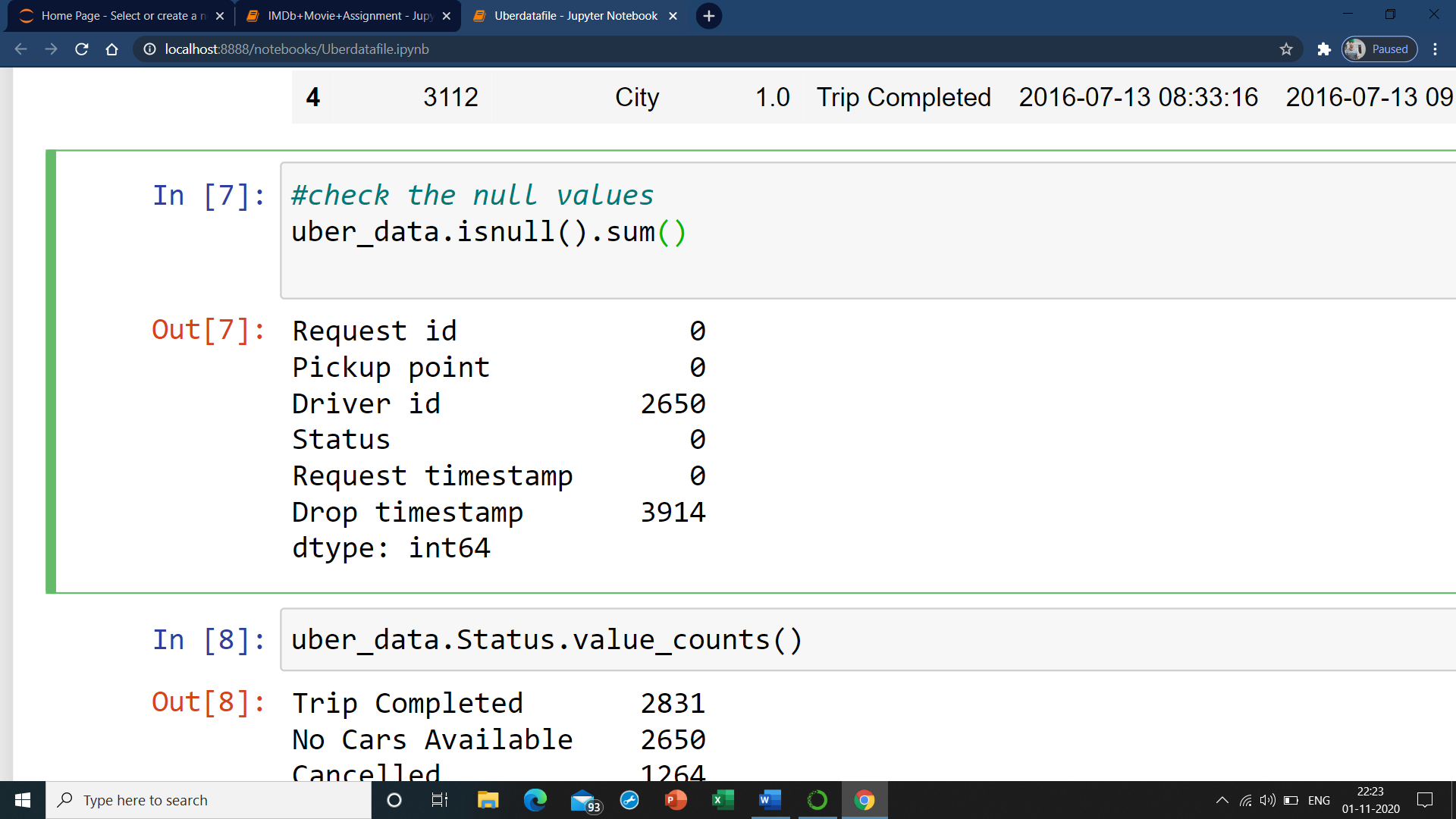
uber\_data["Drop timestamp"] = uber\_data["Drop timestamp"].apply(lambda x: pd.to\_datetime(x))

uber\_data.head()

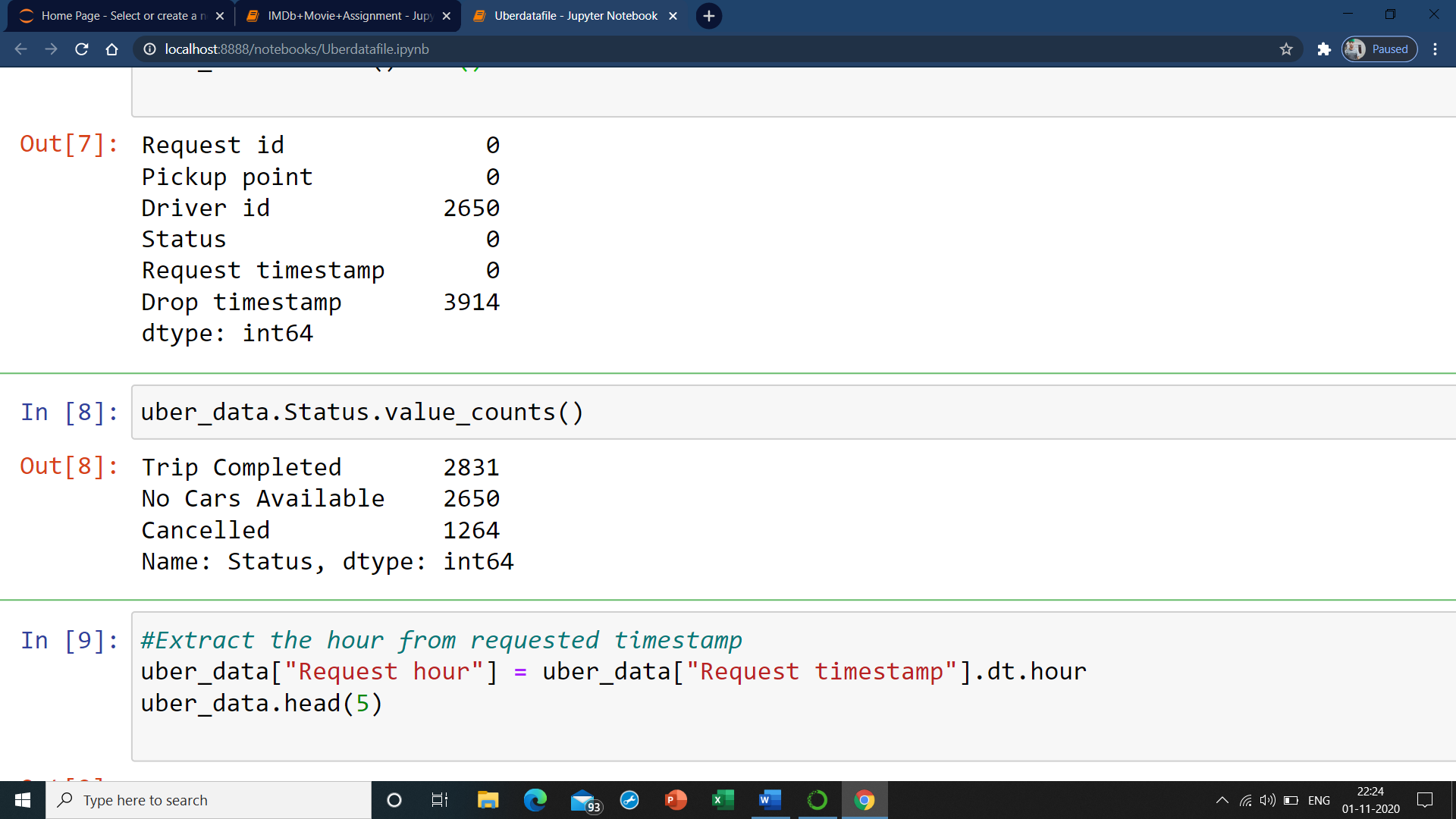


**#check the null values**

uber\_data.isnull().sum()



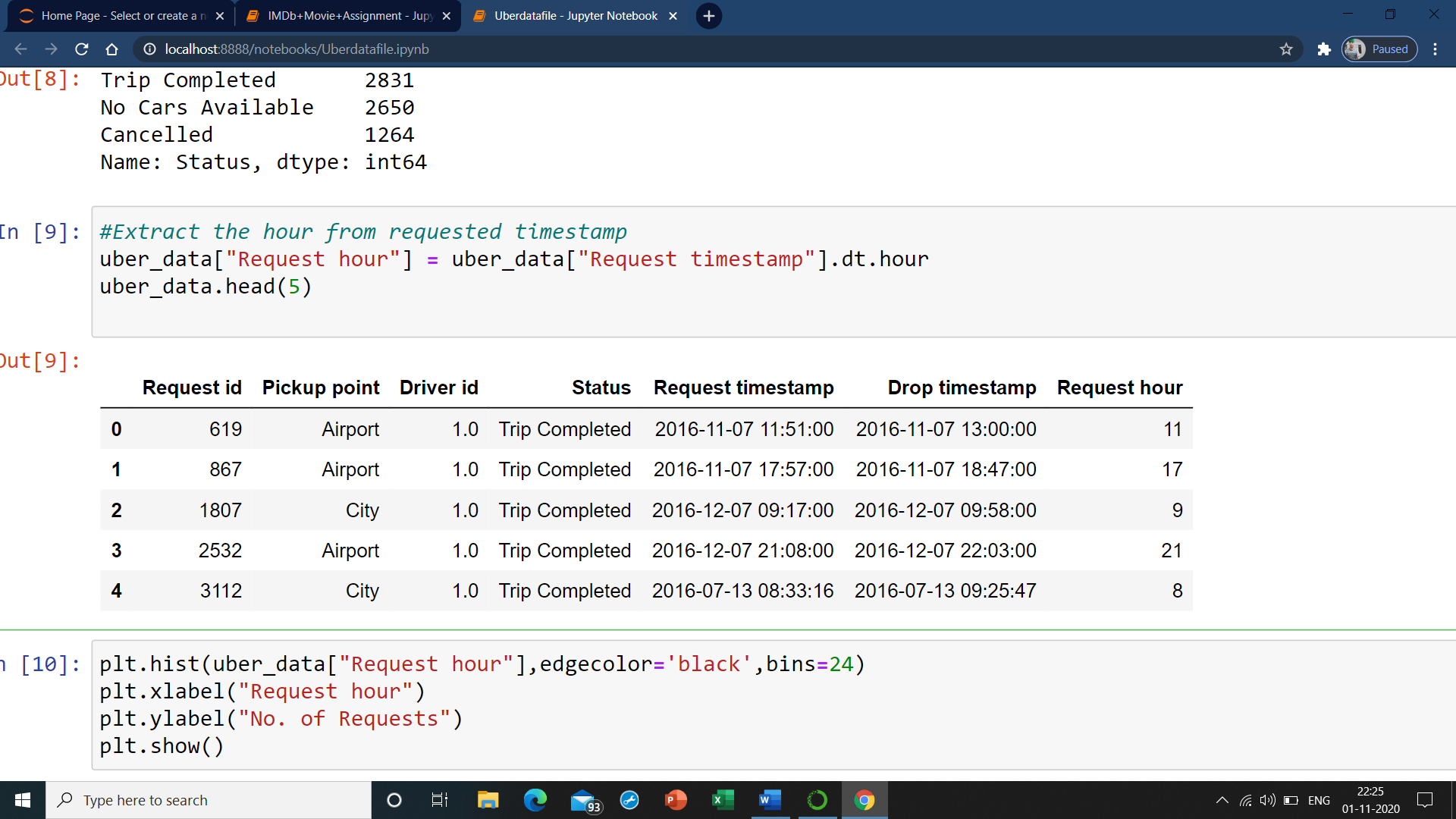
**uber\_data.Status.value\_counts()**



**#Extract the hour from requested timestamp**

uber\_data["Request hour"] = uber\_data["Request timestamp"].dt.hour

uber\_data.head(5)

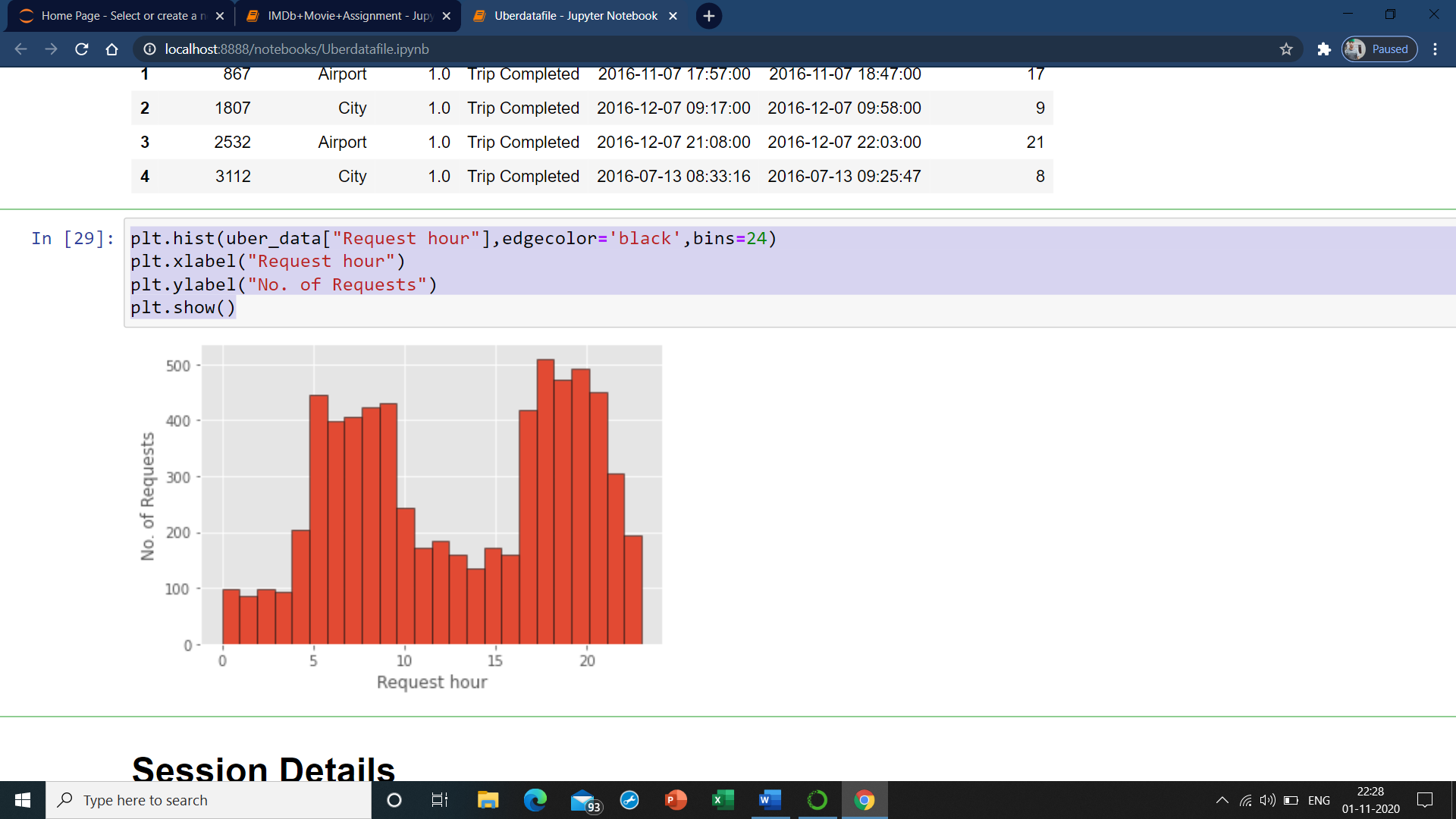


**plt.hist(uber\_data["Request hour"],edgecolor='black',bins=24)**

**plt.xlabel("Request hour")**

**plt.ylabel("No. of Requests")**

**plt.show()**



# Session Details

1) EARLY MORNING = Midnight to 5AM

2) MORNING = 5AM to 10AM

3) DAY TIME = 10AM to 5PM

4) EVENING = 5PM to 10PM

5) LATE NIGHT = 10PM TO Midnight

**#Divide the time of the day into five categories**

def time\_period(x):

if x < 5:

return "Early Morning"

elif 5 <= x < 10:

return "Morning"

elif 10 <= x < 17:

return "Day Time"

elif 17 <= x < 22:

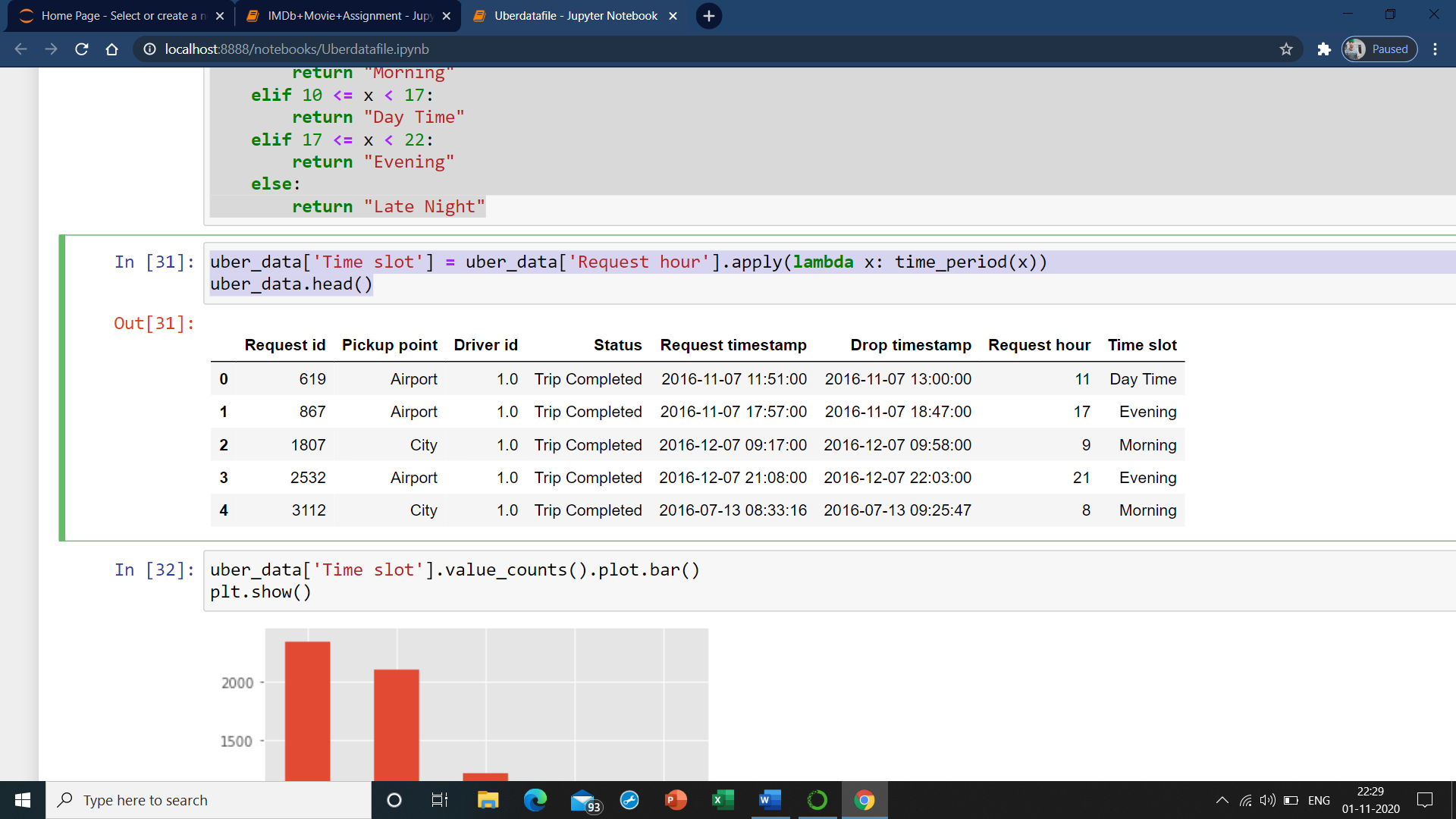
return "Evening"

else:

return "Late Night**"**

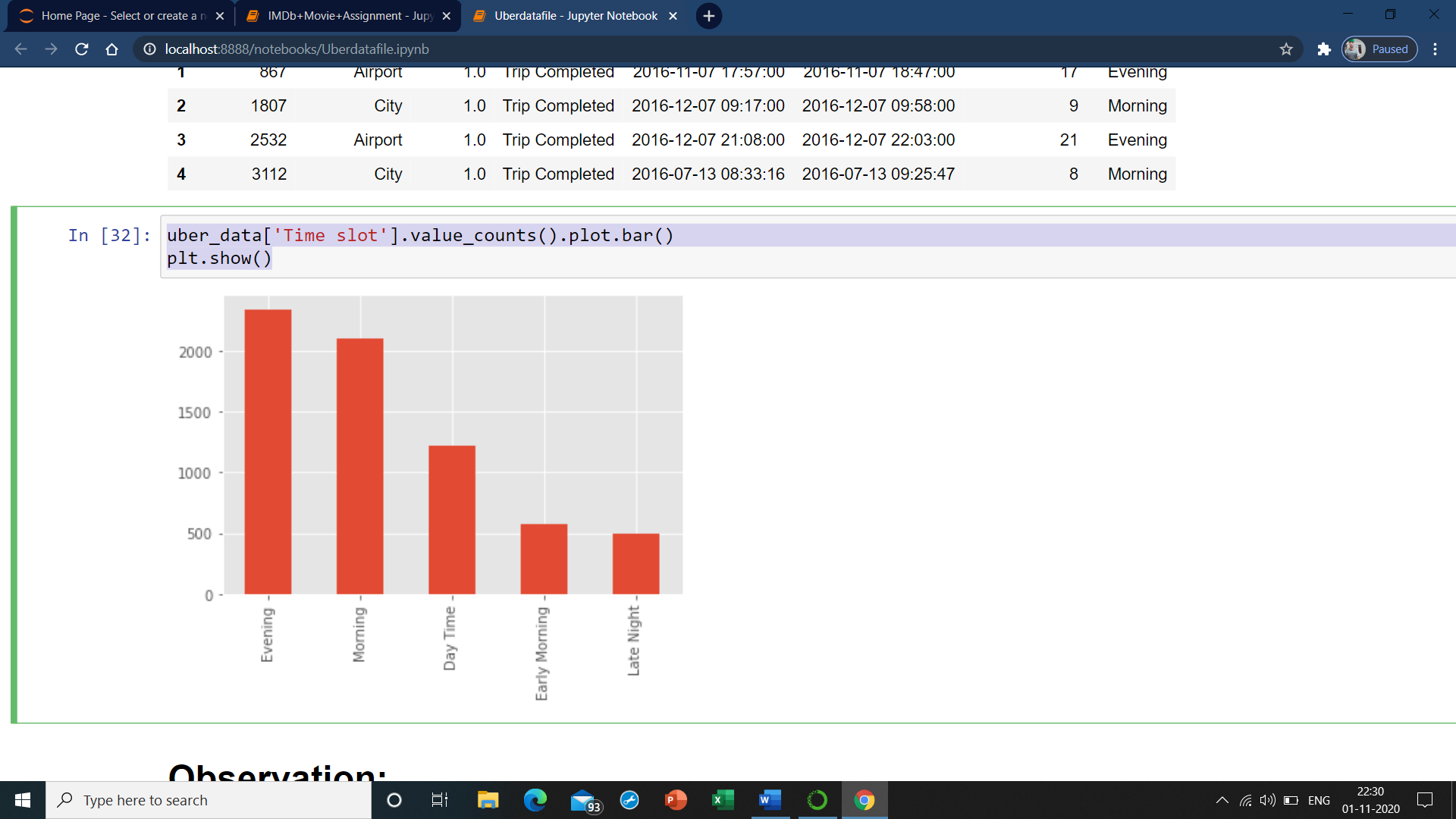
uber\_data['Time slot'] = uber\_data['Request hour'].apply(lambda x: time\_period(x))

uber\_data.head()



**uber\_data['Time slot'].value\_counts().plot.bar()**

**plt.show()**

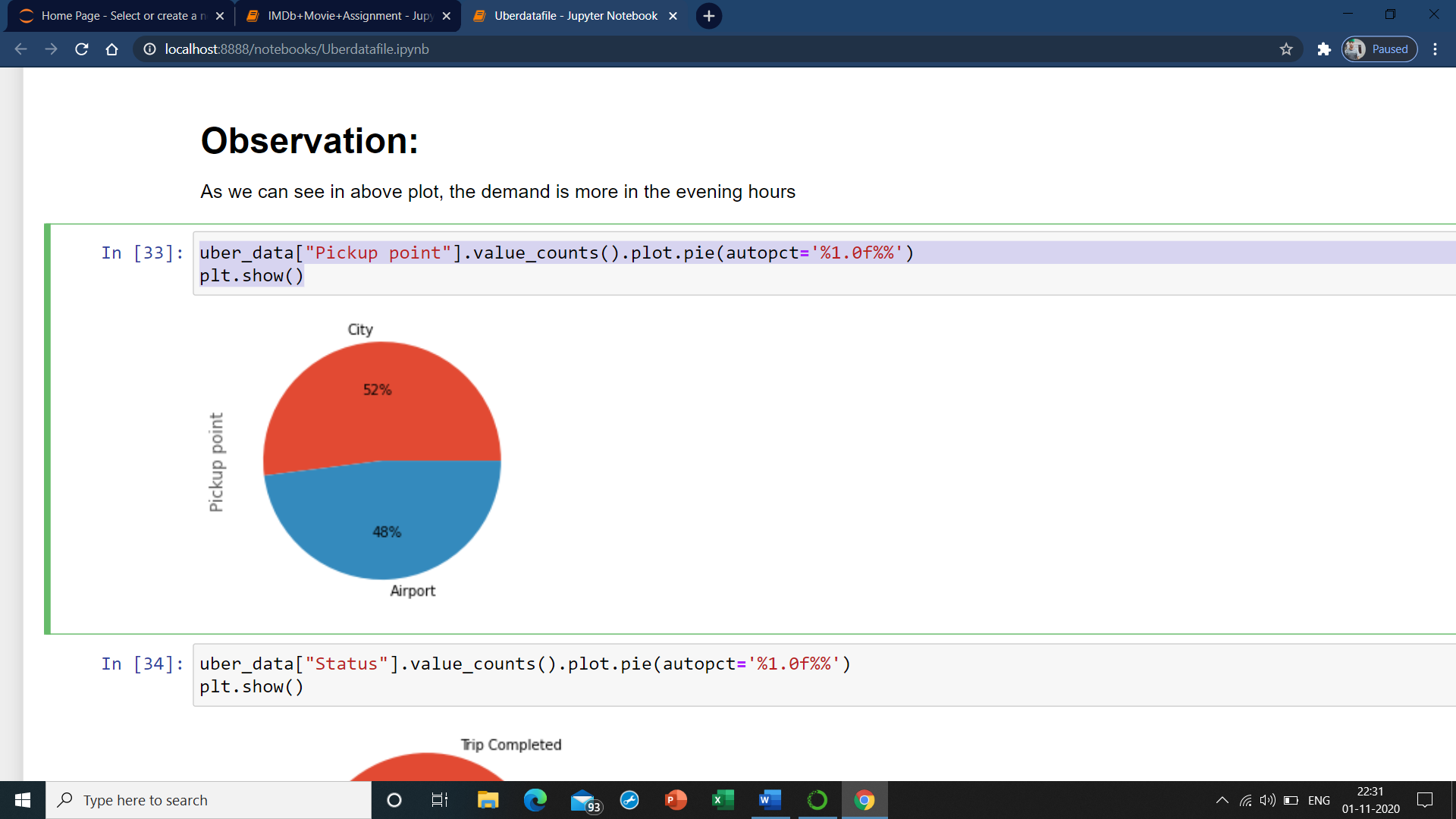


# Observation:

As we can see in above plot, the demand is more in the evening hours

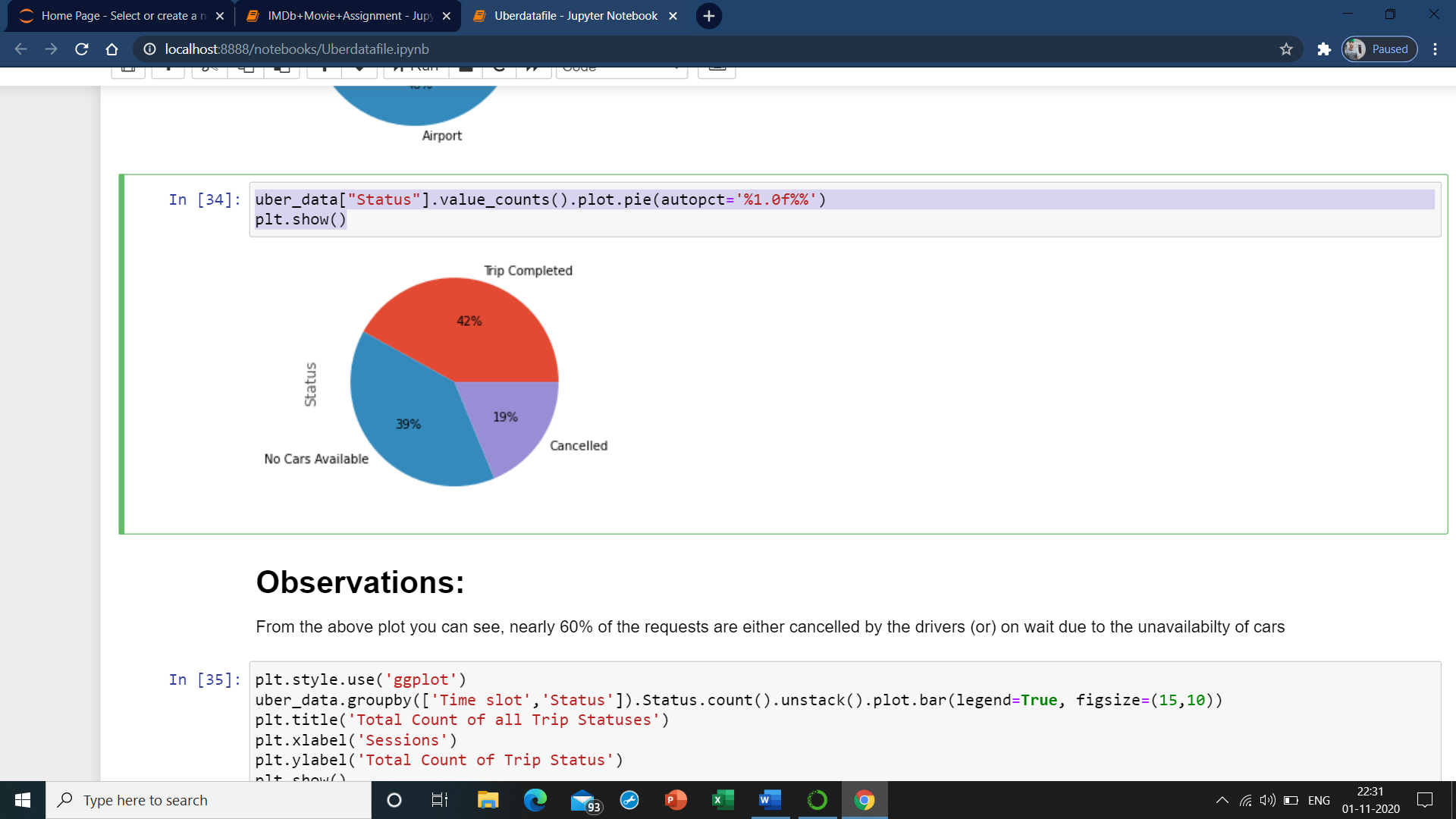
**uber\_data["Pickup point"].value\_counts().plot.pie(autopct='%1.0f%%')**

**plt.show()**



**uber\_data["Status"].value\_counts().plot.pie(autopct='%1.0f%%')**

**plt.show()**



# Observations:

From the above plot you can see, nearly 60% of the requests are either cancelled by the drivers (or) on wait due to the unavailabilty of cars

plt.style.use('ggplot')

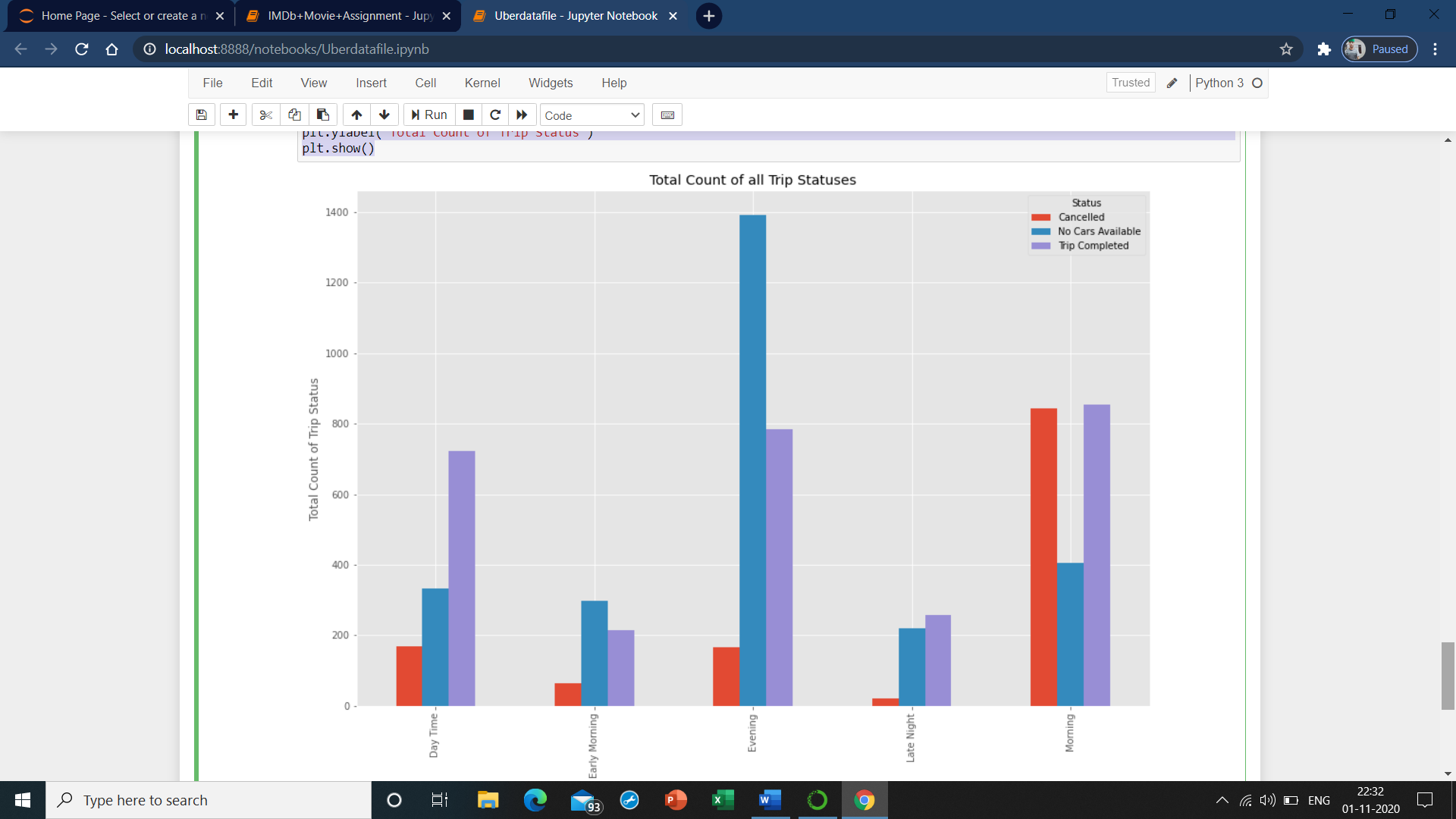
uber\_data.groupby(['Time slot','Status']).Status.count().unstack().plot.bar(legend=True, figsize=(15,10))

plt.title('Total Count of all Trip Statuses')

plt.xlabel('Sessions')

plt.ylabel('Total Count of Trip Status')

plt.show()



# Observations:

1) Maximum number of cancellations are being done during morning hours from 5AM to 10AM by the drivers, this happens mainly due to less demand for the cabs from airport to city. This might be due to few number of flight arrivals at the airport in the morning. So drivers are not willing to take the trip as they will not have a booking to return back to the city, hence they cancel the trip.

2) Customers find massive number of unavailable cars during evening hours from 5PM to 10PM, this could be due to huge number of flight arrivals and departures in the evening, that results in high demand for the cabs. Hence customers could not find a cab in the evening hours.

**POSSIBLE SOLUTION:**

1) As we observe huge number of cancellations are being done during morning hours, I suggest uber management to provide some bonus for each trip and also some incentives for drivers during this hours

2) Uber can also put some offers for the customers during late nights where demand is low and if possible increase the number of cabs during busy hours which would be beneficial