```
! python -m pip install 'fsspec>=0.3.3'
import dask.dataframe as dd
from dask.diagnostics import ProgressBar
import requests

     Requirement already satisfied: fsspec>=0.3.3 in /usr/local/lib/python3.10/dist-packages (2023.6.0)
```

```
 # python -m pip install dask[dataframe] --upgrade  # or python -m pip install
 ! pip install dask[dataframe]

     Requirement already satisfied: dask[dataframe] in /usr/local/lib/python3.10/dist-packages (2023.8.1)
     Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-packages (from dask[dataframe]) (8.1.7)
     Requirement already satisfied: cloudpickle>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from dask[dataframe]) (2.2.1)
     Requirement already satisfied: fsspec>=2021.09.0 in /usr/local/lib/python3.10/dist-packages (from dask[dataframe]) (2023.6.0)
     Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from dask[dataframe]) (24.0)
     Requirement already satisfied: partd>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from dask[dataframe]) (1.4.1)
     Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from dask[dataframe]) (6.0.1)
     Requirement already satisfied: toolz>=0.10.0 in /usr/local/lib/python3.10/dist-packages (from dask[dataframe]) (0.12.1)
     Requirement already satisfied: importlib-metadata>=4.13.0 in /usr/local/lib/python3.10/dist-packages (from dask[dataframe]) (7.1.0)
     Requirement already satisfied: pandas>=1.3 in /usr/local/lib/python3.10/dist-packages (from dask[dataframe]) (1.5.3)
     Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata>=4.13.0->dask[dataframe]) (3.18.1)
     Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3->dask[dataframe]) (2.8.2)
     Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3->dask[dataframe]) (2023.4)
     Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3->dask[dataframe]) (1.25.2)
     Requirement already satisfied: locket in /usr/local/lib/python3.10/dist-packages (from partd>=1.2.0->dask[dataframe]) (1.0.0)
     Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas>=1.3->dask[dataframe]) (1.16.0)
```

```
! pip install dask

     Requirement already satisfied: dask in /usr/local/lib/python3.10/dist-packages (2023.8.1)
     Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-packages (from dask) (8.1.7)
     Requirement already satisfied: cloudpickle>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from dask) (2.2.1)
     Requirement already satisfied: fsspec>=2021.09.0 in /usr/local/lib/python3.10/dist-packages (from dask) (2023.6.0)
     Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from dask) (24.0)
     Requirement already satisfied: partd>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from dask) (1.4.1)
     Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from dask) (6.0.1)
     Requirement already satisfied: toolz>=0.10.0 in /usr/local/lib/python3.10/dist-packages (from dask) (0.12.1)
     Requirement already satisfied: importlib-metadata>=4.13.0 in /usr/local/lib/python3.10/dist-packages (from dask) (7.1.0)
     Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-packages (from importlib-metadata>=4.13.0->dask) (3.18.1)
     Requirement already satisfied: locket in /usr/local/lib/python3.10/dist-packages (from partd>=1.2.0->dask) (1.0.0)
```

```
! pip install requests
! pip install aiohttp
! pip install pandas

     Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (2.31.0)
     Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests) (3.3.2)
     Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests) (3.6)
     Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests) (2.0.7)
     Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests) (2024.2.2)
     Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (3.9.3)
     Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp) (1.3.1)
     Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp) (23.2.0)
     Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp) (1.4.1)
     Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp) (6.0.5)
     Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp) (1.9.4)
     Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp) (4.0.3)
     Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.10/dist-packages (from yarl<2.0,>=1.0->aiohttp) (3.6)
     Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (1.5.3)
     Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
     Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
     Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.25.2)
     Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

```
import dask.dataframe as dd
import pandas as pd
from dask.diagnostics import ProgressBar
from matplotlib import pyplot as plt
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import drive
drive.mount('/content/drive')

     Mounted at /content/drive
```

```
cd '/content/drive/MyDrive/Colab Notebooks/'

     /content/drive/MyDrive/Colab Notebooks
```

```
#Load CSV using dask Method
import dask.dataframe as dd

#Load CSV using Dask
df = dd.read_csv('car_prices.csv')
df
```

Dask DataFrame Structure:

	year	make	model	trim	body	transmission	vin	state	condition	odometer (mileage)	color	interior	seller	mmr	sellingprice	saledate
npartitions=1																
	int64	object	object	object	object	object	object	object	float64	int64	object	object	object	int64	int64	ob
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Dask Name: read-csv, 1 graph layer

```
#Check the shape of data
print("Shape of the dataset:", df.shape[0], "rows and", len(df.columns), "columns")
```

```
Shape of the dataset: Delayed('int-267b6dc6-c8f6-4073-908b-d319bc030ec5') rows and 16 columns
```

```
#Check data types of data
print("Data types of columns:")
print(df.dtypes)
```

```
Data types of columns:
year                   int64
make                  object
model                 object
trim                  object
body                  object
transmission          object
vin                   object
state                 object
condition            float64
odometer (mileage)     int64
color                 object
interior              object
seller                object
mmr                    int64
sellingprice           int64
saledate              object
dtype: object
```

```
import dask.dataframe as dd

# Load CSV using Dask, specifying the dtype for "odometer (mileage)"
df = dd.read_csv('car_prices.csv', dtype={'odometer (mileage)': 'float64'})

# Now you can proceed with printing the top 20 rows or other operations
print("Top 20 rows:")
print(df.head(20))
```

```
2   Thu Jan 15 2015 04:30:00 GMT-0800 (PST)
3   Thu Jan 29 2015 04:30:00 GMT-0800 (PST)
4   Thu Dec 18 2014 12:30:00 GMT-0800 (PST)
5   Tue Dec 30 2014 12:00:00 GMT-0800 (PST)
6   Wed Dec 17 2014 12:30:00 GMT-0800 (PST)
7   Tue Dec 16 2014 13:00:00 GMT-0800 (PST)
8   Thu Dec 18 2014 12:00:00 GMT-0800 (PST)
9   Tue Jan 20 2015 04:00:00 GMT-0800 (PST)
10  Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
11  Tue Dec 16 2014 12:00:00 GMT-0800 (PST)
12  Tue Jan 13 2015 12:00:00 GMT-0800 (PST)
13  Tue Dec 16 2014 12:30:00 GMT-0800 (PST)
14  Tue Dec 16 2014 12:00:00 GMT-0800 (PST)
15  Tue Dec 23 2014 12:00:00 GMT-0800 (PST)
16  Tue Dec 16 2014 13:00:00 GMT-0800 (PST)
17  Thu Dec 18 2014 12:30:00 GMT-0800 (PST)
18  Tue Dec 30 2014 15:00:00 GMT-0800 (PST)
19  Wed Dec 17 2014 12:30:00 GMT-0800 (PST)
```

Double-click (or enter) to edit

```
#Display bottom 20 rows
print("Bottom 20 rows:")
print(df.tail(20))
```

```
Bottom 20 rows:
        year           make           model                   trim  \
558817  2012           Ford            Flex                    SEL   
558818  2013      Chevrolet  Silverado 1500                     LT   
558819  2012            Kia          Optima                     EX   
558820  2014          Dodge         Charger                     SE   
558821  2012           Ford          Escape                    XLT   
558822  2009  Mercedes-Benz         C-Class            C300 Luxury   
558823  2012      Chevrolet  Silverado 1500                     LT   
558824  2013           Audi              S5    Premium Plus quattro   
558825  2011         Subaru        Forester                   2.5X   
558826  2014           Jeep  Grand Cherokee                Limited   
558827  2014           Jeep  Grand Cherokee                 Laredo   
558828  2012          Dodge   Grand Caravan  American Value Package   
558829  2012        Hyundai         Elantra                Limited   
558830  2012         Nissan          Sentra                 2.0 SR   
558831  2011            BMW        5 Series                   528i   
558832  2015            Kia            K900                 Luxury   
558833  2012            Ram            2500            Power Wagon   
558834  2012            BMW              X5              xDrive35d   
558835  2015         Nissan          Altima                  2.5 S   
558836  2014           Ford           F-150                    XLT   

               body transmission                vin state  condition  \
558817        Wagon    automatic  2fmhk6cc1cbd17905    ny        3.9   
558818     crew cab    automatic  3gcpcse0xdg244430    tx        4.3   
558819        Sedan    automatic  5xxgn4a74cg032147    fl        4.4   
558820        Sedan          NaN  2c3cdxbg9eh324236    va        4.2   
558821          SUV    automatic  1fmcu9d78ckc84074    fl        3.8   
558822        sedan    automatic  wddgf54x89r068689    hi        4.1   
558823     Crew Cab    automatic  3gcpcse00cg289987    tx        3.7   
558824  convertible    automatic  waucgafh6dn005382    fl        5.0   
558825          suv       manual  jf2shbac9bg741815    ca        4.1   
558826          SUV    automatic  1c4rjebg4ec573100    ca        4.4   
558827          SUV    automatic  1c4rjfag0ec466276    pa        4.2   
558828      Minivan    automatic  2c4rdgbg1cr349287    ma        3.7   
558829        Sedan          NaN  5npdh4ae7ch106397    pa        4.0   
558830        Sedan          NaN  3n1ab6ap3cl622485    tn        2.6   
558831        Sedan    automatic  wbafr1c53bc744672    fl        3.9   
558832        Sedan          NaN  knalw4d4xf6019304    in        4.5   
558833     Crew Cab    automatic  3c6td5et6cg112407    wa        5.0   
558834          SUV    automatic  5uxzw0c58cl668465    ca        4.8   
558835        sedan    automatic  1n4al3ap0fc216050    ga        3.8   
558836    SuperCrew    automatic  1ftfw1et2eke87277    ca        3.4   

        odometer (mileage)   color interior  \
558817             28320.0     red    black   
558818             74575.0   black    black   
558819             58176.0     red    beige   
558820             22744.0   white    black   
558821             74673.0   white     gray   
558822             80498.0  silver    black   
558823             37908.0   white    black   
558824             20158.0  silver    black   
558825             71693.0  silver    black   
558826              9024.0    gray    black   
558827             25180.0    gray    black   
558828             97036.0  silver     gray   
```

```
# https://data.cityofnewyork.us/browse?q=parking+ticket
# 2016 https://data.cityofnewyork.us/resource/kiv2-tbus.csv
# 2015 https://data.cityofnewyork.us/resource/c284-tqph.csv
# 2014 https://data.cityofnewyork.us/resource/jt7v-77mi.csv
# Hypothesis 1: Car price correlates positively with the number of cylinders.

    #Null Hypothesis (H0): There is no correlation between car price and the number of cylinders.
    #Alternate Hypothesis (H1): There is a positive correlation between car price and the number of cylinders.

# Hypothesis 2: Cars with higher horsepower tend to have higher prices.

    #Null Hypothesis (H0): There is no correlation between car price and horsepower.
    #Alternate Hypothesis (H1): There is a positive correlation between car price and horsepower.

# Hypothesis 3: Fuel efficiency (mpg) negatively correlates with price.

    #Null Hypothesis (H0): There is no correlation between car price and fuel efficiency (mpg).
    #Alternate Hypothesis (H1): There is a negative correlation between car price and fuel efficiency (mpg).

# Hypothesis 4: Cars from certain brands (e.g., luxury brands) have higher average prices.

    #Null Hypothesis (H0): There is no difference in average prices between luxury and non-luxury car brands.
    #Alternate Hypothesis (H1): Luxury car brands have a higher average price compared to non-luxury brands.

# Hypothesis 5: The age of the car (model year) inversely affects the price.

    #Null Hypothesis (H0): There is no correlation between car price and model year.
    #Alternate Hypothesis (H1): There is a negative correlation between car price and model year (older cars are cheaper).


# Prediction:
#I would suggest a prediction experiment focusing on car price prediction. Here's why:

#Reasons for Prediction:    #Continuous Target Variable: The target variable you're interested in, "sellingprice," is continuous (numerical values).
#Classification is typically used for categorical target variables (e.g., predicting if a car is "luxury" or "non-luxury").
#Prediction is better suited for estimating continuous values like price.

#Granular Insights: Predicting the actual selling price provides more granular and actionable insights than simply classifying cars into categories. Knowing the esti

#Example: #Imagine you're building a model to help car sellers determine an appropriate selling price.
```

Common columns