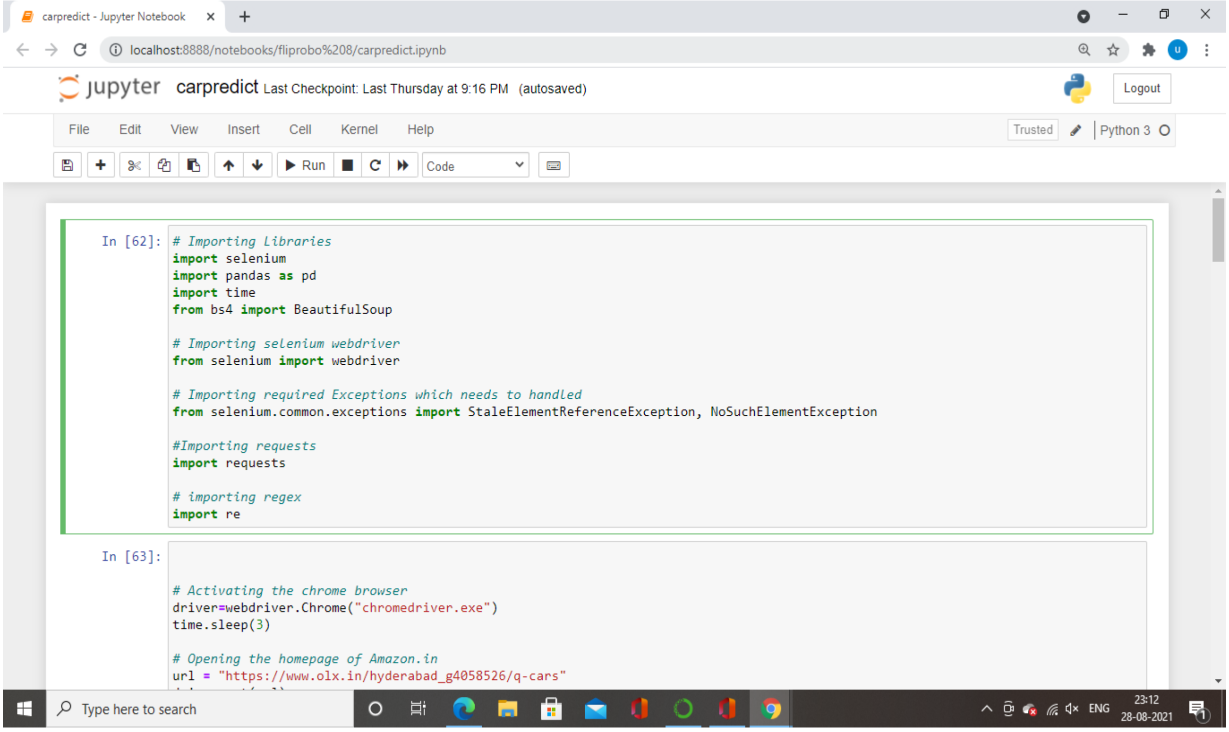
Car Price Prediction Project

covid 19 impact in the market, we have seen lot of changes in the car market. Now some cars are in demand hence making them costly and some are not in demand hence cheaper. One of our clients works with small traders, who sell used cars. With the change in market due to covid 19 impact, our client is facing problems with their previous car price valuation machine learning models.

The number of columns for data doesn't have limit, it's up to you and your creativity. Generally, these columns areBrand, model, variant, manufacturing year, driven kilometers, fuel, number of owners, location andat last target variable Price of the car. This data is to give you a hint about important variables in used car model.

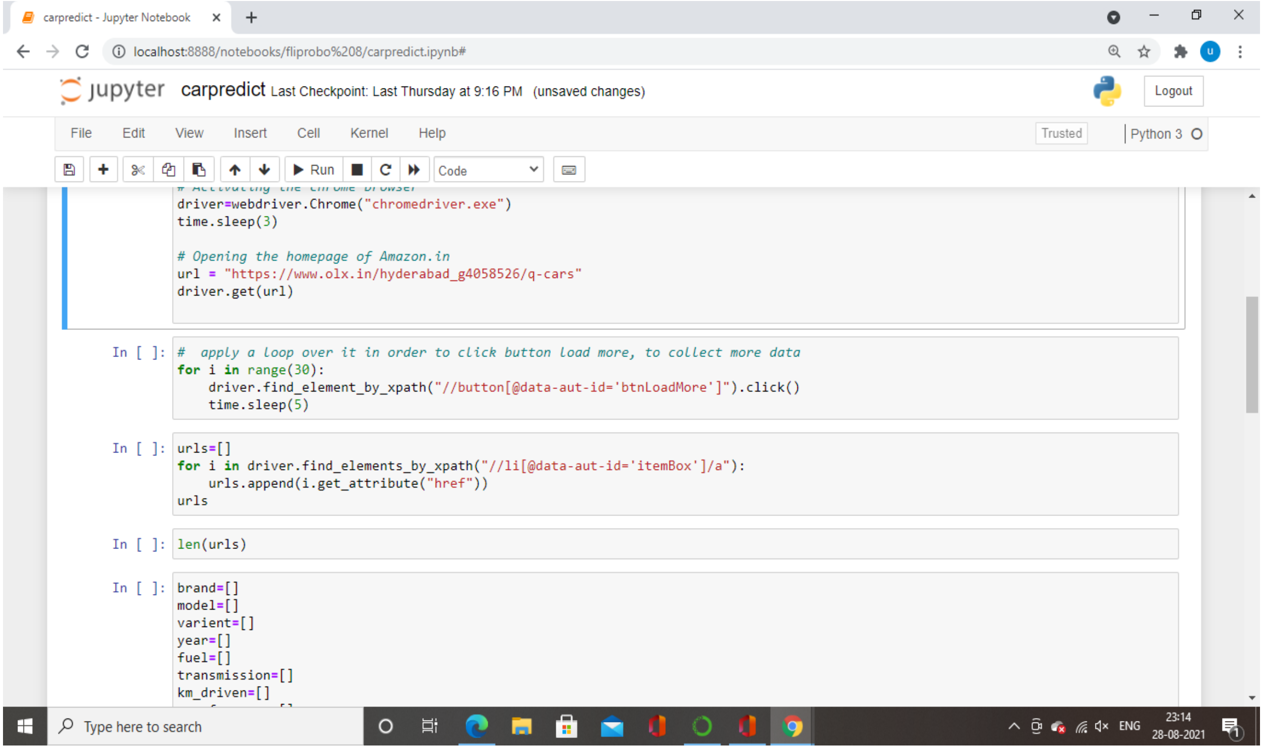
Web scraping


carpredict - Jupyter Notebook
localhost:8888/notebooks/fliprobo%208/carpredict.ipynb
jupyter carpredict Last Checkpoint: Last Thursday at 9:16 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
23:12
28-08-2021

```
In [62]: # Importing Libraries
         import selenium
         import pandas as pd
         import time
         from bs4 import BeautifulSoup

         # Importing selenium webdriver
         from selenium import webdriver

         # Importing required Exceptions which needs to handled
         from selenium.common.exceptions import StaleElementReferenceException, NoSuchElementException

         #Importing requests
         import requests

         # importing regex
         import re

In [63]: 

         # Activating the chrome browser
         driver=webdriver.Chrome("chromedriver.exe")
         time.sleep(3)

         # Opening the homepage of Amazon.in
         url = "https://www.olx.in/hyderabad_g4058526/q-cars"
```

By importing libraries selenium we are going to scrape the data from website and also we are

Using the exception handling


carpredict - Jupyter Notebook
localhost:8888/notebooks/fliprobo%208/carpredict.ipynb#
jupyter carpredict Last Checkpoint: Last Thursday at 9:16 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
Type here to search
ENG
23:14
28-08-2021

```
# Activating the chrome browser
driver=webdriver.Chrome("chromedriver.exe")
time.sleep(3)

# Opening the homepage of Amazon.in
url = "https://www.olx.in/hyderabad_g4058526/q-cars"
driver.get(url)
```

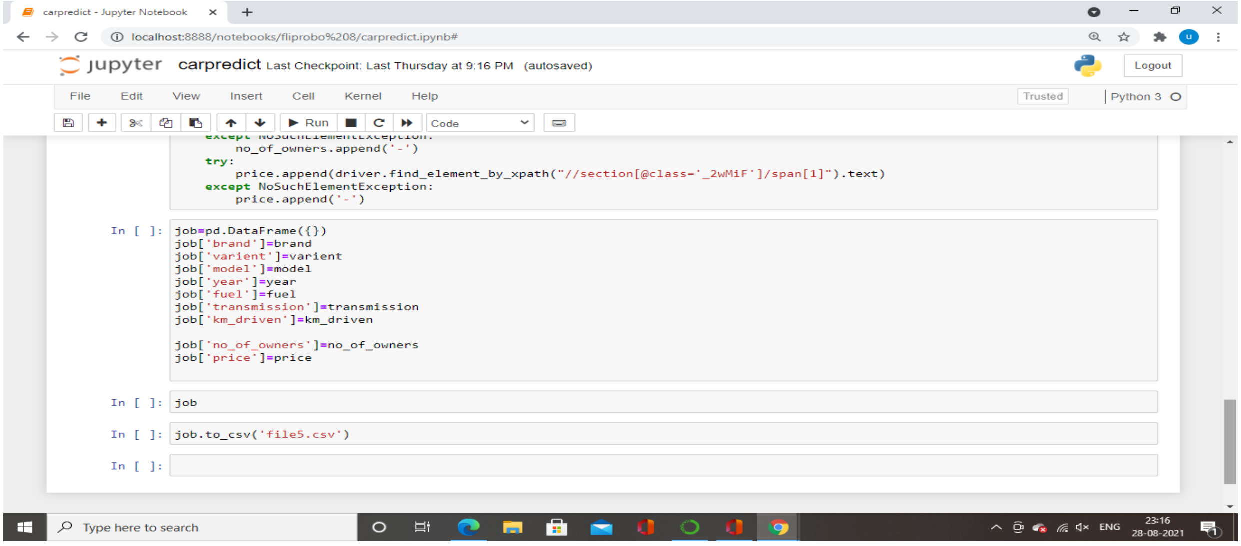
```
In [ ]: #  apply a loop over it in order to click button load more, to collect more data
        for i in range(30):
            driver.find_element_by_xpath("//button[@data-aut-id='btnLoadMore']").click()
            time.sleep(5)
```

```
In [ ]: urls=[]
        for i in driver.find_elements_by_xpath("//li[@data-aut-id='itemBox']/a"):
            urls.append(i.get_attribute("href"))
        urls
```

```
In [ ]: len(urls)
```

```
In [ ]: brand=[]
        model=[]
        varient=[]
        year=[]
        fuel=[]
        transmission=[]
        km_driven=[]
```

By using url we are going to collect data from websites


carpredict - Jupyter Notebook
localhost:8888/notebooks/fliprobo%208/carpredict.ipynb#
jupyter carpredict Last Checkpoint: Last Thursday at 9:16 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
Type here to search
ENG
23:16
28-08-2021

```
    except NoSuchElementException:
        no_of_owners.append('-')
    try:
        price.append(driver.find_element_by_xpath("//section[@class='_2wMiF']/span[1]").text)
    except NoSuchElementException:
        price.append('-')
```

```
In [ ]: job=pd.DataFrame({})
        job['brand']=brand
        job['varient']=varient
        job['model']=model
        job['year']=year
        job['fuel']=fuel
        job['transmission']=transmission
        job['km_driven']=km_driven

        job['no_of_owners']=no_of_owners
        job['price']=price
```

```
In [ ]: job
```

```
In [ ]: job.to_csv('file5.csv')
```

```
In [ ]:
```

By using data frame we are saving data

Machine learning model

We using library numpy,pandas matplotlib,seaborn

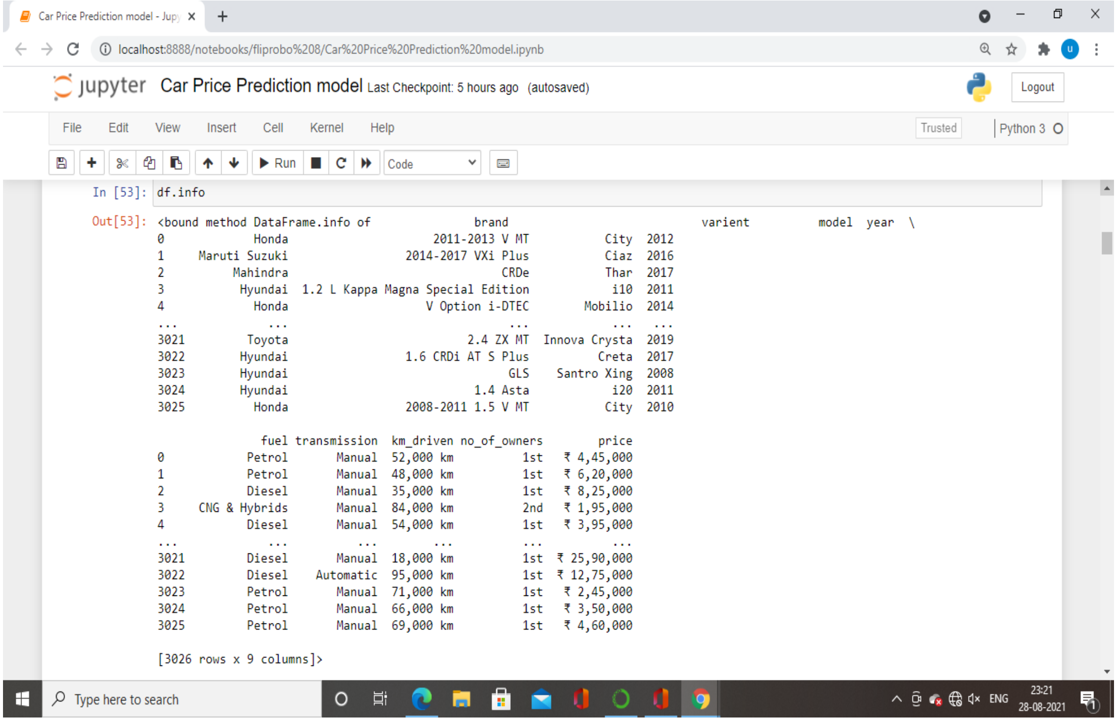
We are reading the data

#unnamed column droped

df=df.drop(['Unnamed: 0'], axis = 1)

We are going to drop column

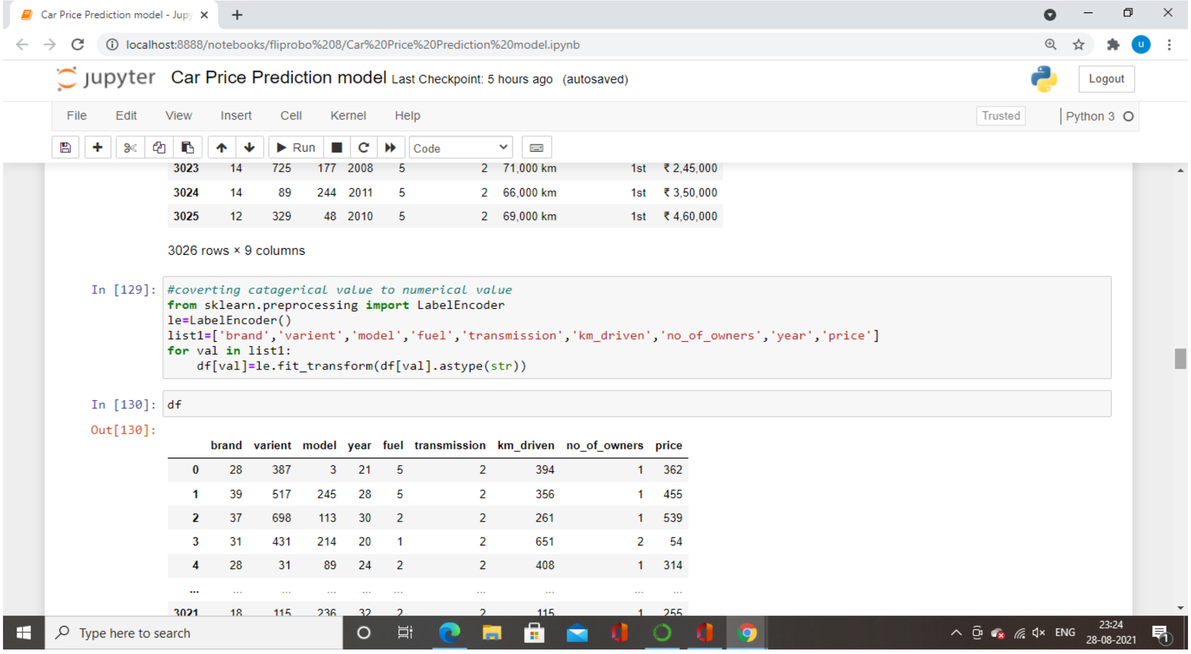
Df.shape we are going to check shape



Car Price Prediction model - Jupy
localhost:8888/notebooks/fliprobo%208/Car%20Price%20Prediction%20model.ipynb
jupyter Car Price Prediction model Last Checkpoint: 5 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code
Type here to search
ENG
23:21
28-08-2021

```
In [53]: df.info

Out[53]: <bound method DataFrame.info of              brand                           varient          model  year  \
         0             Honda                  2011-2013 V MT           City  2012
         1     Maruti Suzuki              2014-2017 VXi Plus           Ciaz  2016
         2          Mahindra                            CRDe           Thar  2017
         3           Hyundai  1.2 L Kappa Magna Special Edition        i10  2011
         4             Honda                 V Option i-DTEC        Mobilio  2014
         ...             ...                             ...            ...   ...
         3021         Toyota                       2.4 ZX MT  Innova Crysta  2019
         3022        Hyundai               1.6 CRDi AT S Plus         Creta  2017
         3023        Hyundai                             GLS   Santro Xing  2008
         3024        Hyundai                        1.4 Asta            i20  2011
         3025          Honda              2008-2011 1.5 V MT           City  2010

                     fuel transmission  km_driven no_of_owners        price
         0         Petrol       Manual  52,000 km          1st  ₹ 4,45,000
         1         Petrol       Manual  48,000 km          1st  ₹ 6,20,000
         2         Diesel       Manual  35,000 km          1st  ₹ 8,25,000
         3     CNG & Hybrids    Manual  84,000 km          2nd  ₹ 1,95,000
         4         Diesel       Manual  54,000 km          1st  ₹ 3,95,000
         ...          ...          ...        ...          ...          ...
         3021      Diesel       Manual  18,000 km          1st ₹ 25,90,000
         3022      Diesel    Automatic  95,000 km          1st ₹ 12,75,000
         3023      Petrol       Manual  71,000 km          1st  ₹ 2,45,000
         3024      Petrol       Manual  66,000 km          1st  ₹ 3,50,000
         3025      Petrol       Manual  69,000 km          1st  ₹ 4,60,000

         [3026 rows x 9 columns]>
```

Its shows the information



Car Price Prediction model - Jupy

localhost:8888/notebooks/fliprobo%208/Car%20Price%20Prediction%20model.ipynb

jupyter Car Price Prediction model Last Checkpoint: 5 hours ago (autosaved)

3023	14	725	177	2008	5	2	71,000 km	1st	₹ 2,45,000
3024	14	89	244	2011	5	2	66,000 km	1st	₹ 3,50,000
3025	12	329	48	2010	5	2	69,000 km	1st	₹ 4,60,000

3026 rows × 9 columns

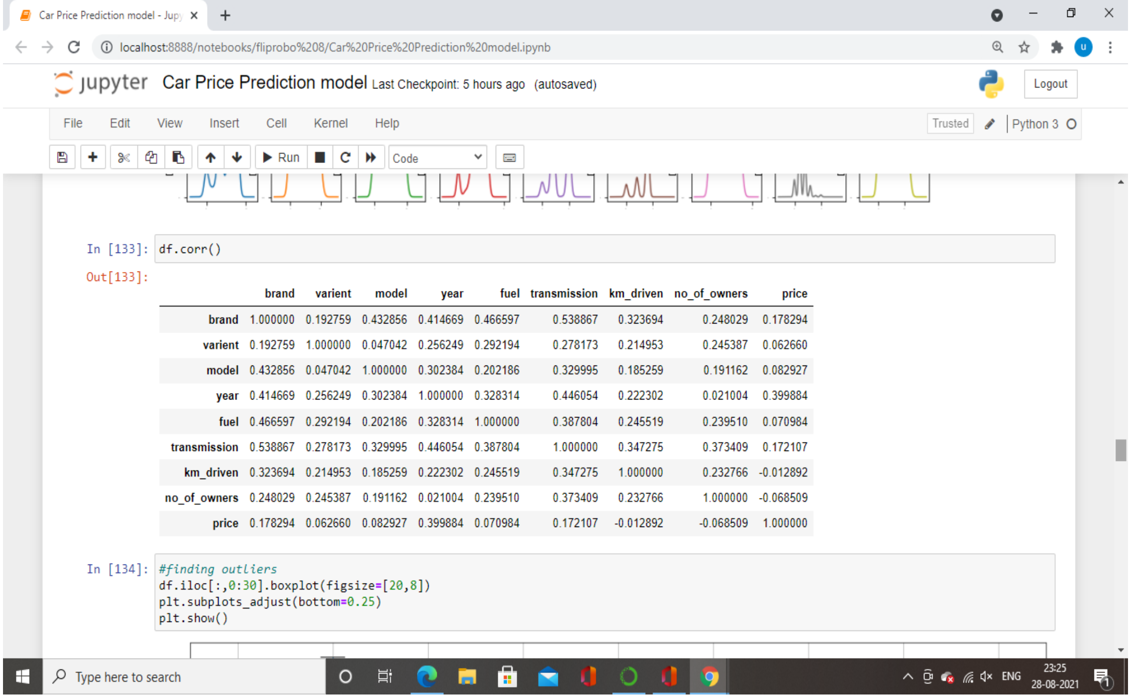
In [129]:
```
#coverting catagerical value to numerical value
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
list1=['brand','varient','model','fuel','transmission','km_driven','no_of_owners','year','price']
for val in list1:
    df[val]=le.fit_transform(df[val].astype(str))
```

In [130]:
```
df
```

Out[130]:

	brand	varient	model	year	fuel	transmission	km_driven	no_of_owners	price
0	28	387	3	21	5	2	394	1	362
1	39	517	245	28	5	2	356	1	455
2	37	698	113	30	2	2	261	1	539
3	31	431	214	20	1	2	651	2	54
4	28	31	89	24	2	2	408	1	314
...	...	...	...	...	...	...	...	...	...
3021	18	115	236	32	2	2	115	1	255

coverting catagerical value to numerical value



Car Price Prediction model - Jupy

localhost:8888/notebooks/fliprobo%208/Car%20Price%20Prediction%20model.ipynb

jupyter Car Price Prediction model Last Checkpoint: 5 hours ago (autosaved)

In [133]:
```
df.corr()
```

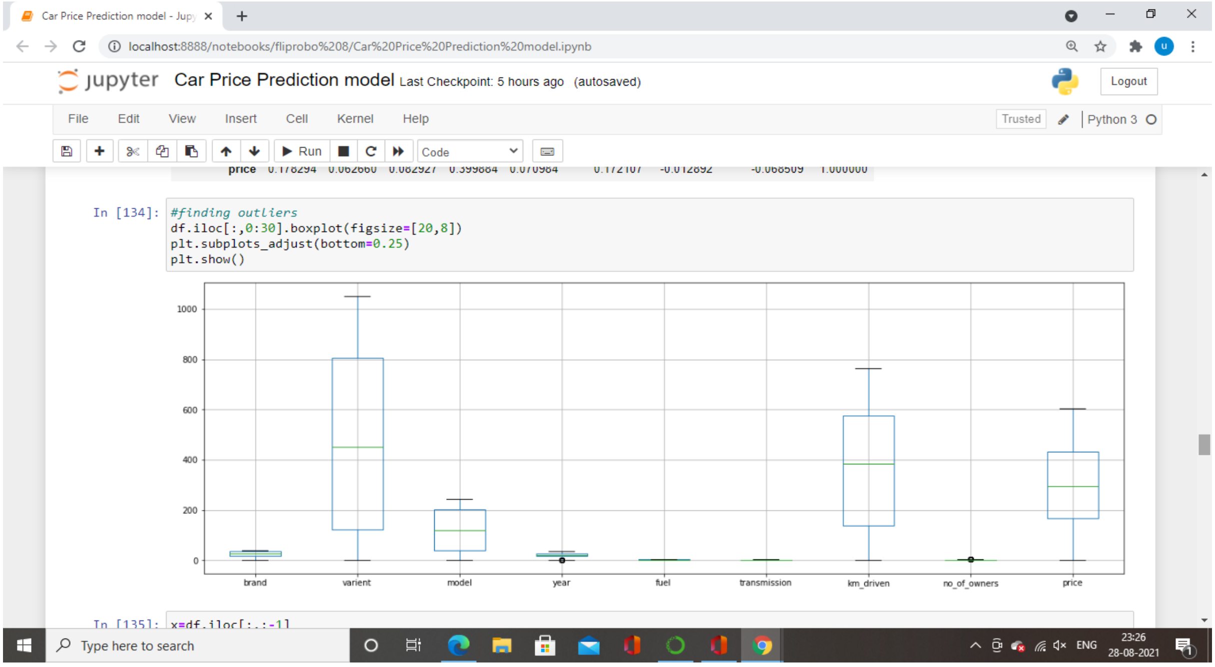
Out[133]:

	brand	varient	model	year	fuel	transmission	km_driven	no_of_owners	price
brand	1.000000	0.192759	0.432856	0.414669	0.466597	0.538867	0.323694	0.248029	0.178294
varient	0.192759	1.000000	0.047042	0.256249	0.292194	0.278173	0.214953	0.245387	0.062660
model	0.432856	0.047042	1.000000	0.302384	0.202186	0.329995	0.185259	0.191162	0.082927
year	0.414669	0.256249	0.302384	1.000000	0.328314	0.446054	0.222302	0.021004	0.399884
fuel	0.466597	0.292194	0.202186	0.328314	1.000000	0.387804	0.245519	0.239510	0.070984
transmission	0.538867	0.278173	0.329995	0.446054	0.387804	1.000000	0.347275	0.373409	0.172107
km_driven	0.323694	0.214953	0.185259	0.222302	0.245519	0.347275	1.000000	0.232766	-0.012892
no_of_owners	0.248029	0.245387	0.191162	0.021004	0.239510	0.373409	0.232766	1.000000	-0.068509
price	0.178294	0.062660	0.082927	0.399884	0.070984	0.172107	-0.012892	-0.068509	1.000000

In [134]:
```
#finding outliers
df.iloc[:,0:30].boxplot(figsize=[20,8])
plt.subplots_adjust(bottom=0.25)
plt.show()
```

Correlelation is find between two variable

Finding the outliers



Car Price Prediction model - Jupy
localhost:8888/notebooks/fliprobo%208/Car%20Price%20Prediction%20model.ipynb
jupyter Car Price Prediction model Last Checkpoint: 5 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3
Run
Code
1000
800
600
400
200
0
brand
varient
model
year
fuel
transmission
km_driven
no_of_owners
price
Type here to search
ENG
23:26
28-08-2021

```
price  0.178294  0.062660  0.082927  0.399884  0.070984     0.172107   -0.012892      -0.068509   1.000000
```

```
In [134]: #finding outliers
          df.iloc[:,0:30].boxplot(figsize=[20,8])
          plt.subplots_adjust(bottom=0.25)
          plt.show()
```

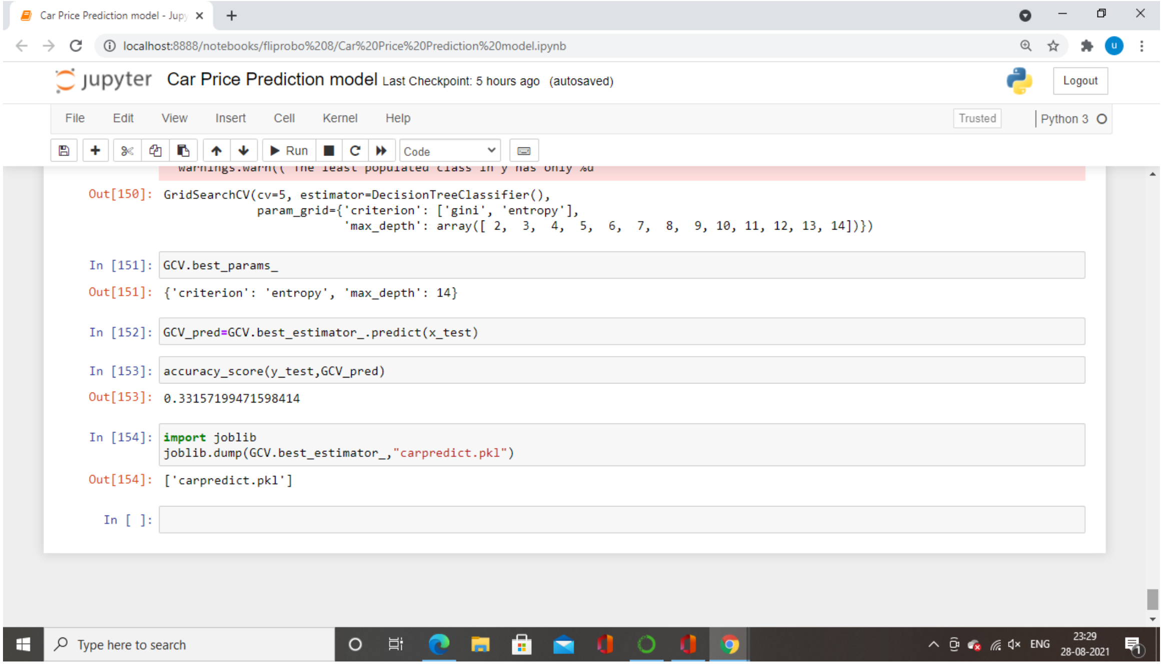
```
In [135]: x=df.iloc[:,:-1]
```

```
Decision Tree Score on Training set is 0.8917907165460381

Decision Tree Score on Test Set is 0.34025138214316

[0.18616299 0.15947788 0.24149528 0.14175681 0.23850803]

Accuracy: 19.35 %

Standard Deviation: 4.05 %

Mean Absolute Error: 80.31780040471207

Mean Squared Error: 20113.017274601996

RMSE: 141.82036974497703

The r2_score is 0.34025138214316
```

```
Random Forest Score on Training set is 0.6804565411013928

Random Forest Score on Test Set is 0.5127518191564149

[0.40356133 0.43719579 0.47227307 0.48417797 0.44626089]

Accuracy: 44.87 %

Standard Deviation: 2.82 %

Mean Absolute Error: 88.44898180179646

Mean Squared Error: 14854.189630832921

RMSE: 121.8777651207673

The r2_score is 0.5127518191564149
```

Cross validation value

```
[-0.2903234   0.83586592  0.83293768 -0.02018227 -0.06782409 -
0.09946589]0.19850132642853868 0.457389134880855
```



Car Price Prediction model - Jupy
localhost:8888/notebooks/fliprobo%208/Car%20Price%20Prediction%20model.ipynb
jupyter
Car Price Prediction model Last Checkpoint: 5 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 3
Run
Code
Type here to search
ENG
23:29
28-08-2021

```
  warnings.warn(( The least populated class in y has only %d

Out[150]: GridSearchCV(cv=5, estimator=DecisionTreeClassifier(),
                       param_grid={'criterion': ['gini', 'entropy'],
                                   'max_depth': array([ 2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])})

In [151]: GCV.best_params_

Out[151]: {'criterion': 'entropy', 'max_depth': 14}

In [152]: GCV_pred=GCV.best_estimator_.predict(x_test)

In [153]: accuracy_score(y_test,GCV_pred)

Out[153]: 0.33157199471598414

In [154]: import joblib
          joblib.dump(GCV.best_estimator_,"carpredict.pkl")

Out[154]: ['carpredict.pkl']

In [ ]:
```