# CAR PRICE PREDICTION

Study of car prices post covid-19 impact

# Problem statement

- With the 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. With the change in market due to covid 19 impact, the previous ML models are not performing well.

- Extracting the car mode, and the cars manufacturer form the name.
- Now name column is done its work and it can be dropped form the dataset.

jupyter model building Last Checkpoint: an hour ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
data = pd.read_csv('cars24data.csv')
```

In [4]:
```
data.head()
```

Out[4]:

| | Unnamed: 0 | Brand | Transmission | Fuel | Kms | Price | year |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 2013 Maruti Swift | Manual | Diesel | 65,561 km | 3,32,699 | 2013 |
| 1 | 1 | 2016 Maruti Swift | Manual | Petrol | 58,818 km | 4,07,599 | 2016 |
| 2 | 2 | 2014 Maruti Swift Dzire | Manual | Diesel | 55,542 km | 3,43,799 | 2014 |
| 3 | 3 | 2018 Maruti Vitara Brezza | Manual | Diesel | 71,192 km | 6,74,099 | 2018 |
| 4 | 4 | 2014 Maruti Swift Dzire | Manual | Diesel | 55,542 km | 3,43,799 | 2014 |

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
Unnamed: 0      0
Brand           0
Transmission    0
Fuel            0
Kms             0
Price           0
year            0
dtype: int64
```

In [8]:
```
data['Brand'].value_counts()
```

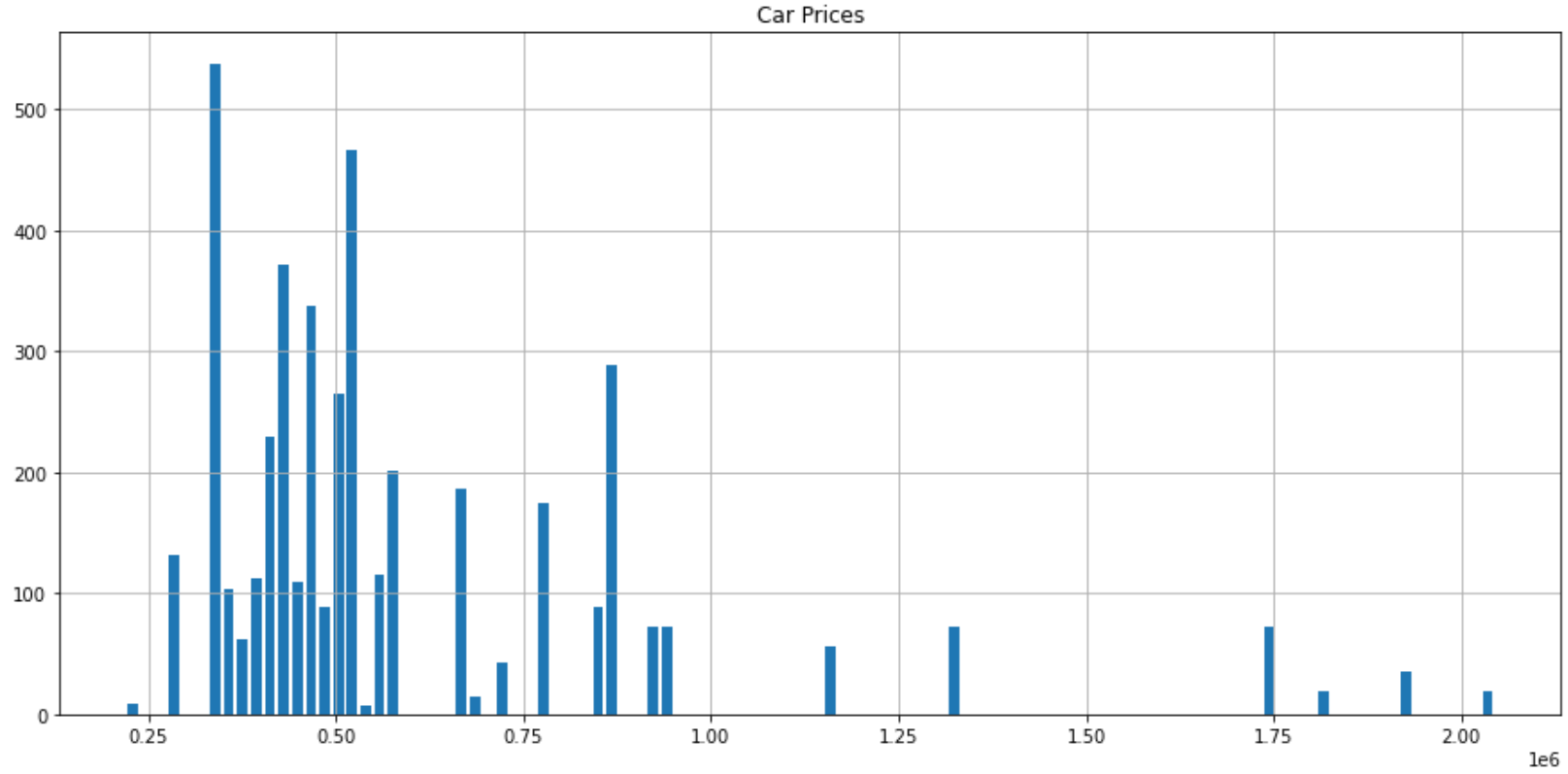
# EDA

Minimum Mileage 350 km; Maximum Mileage 1,000,000 km

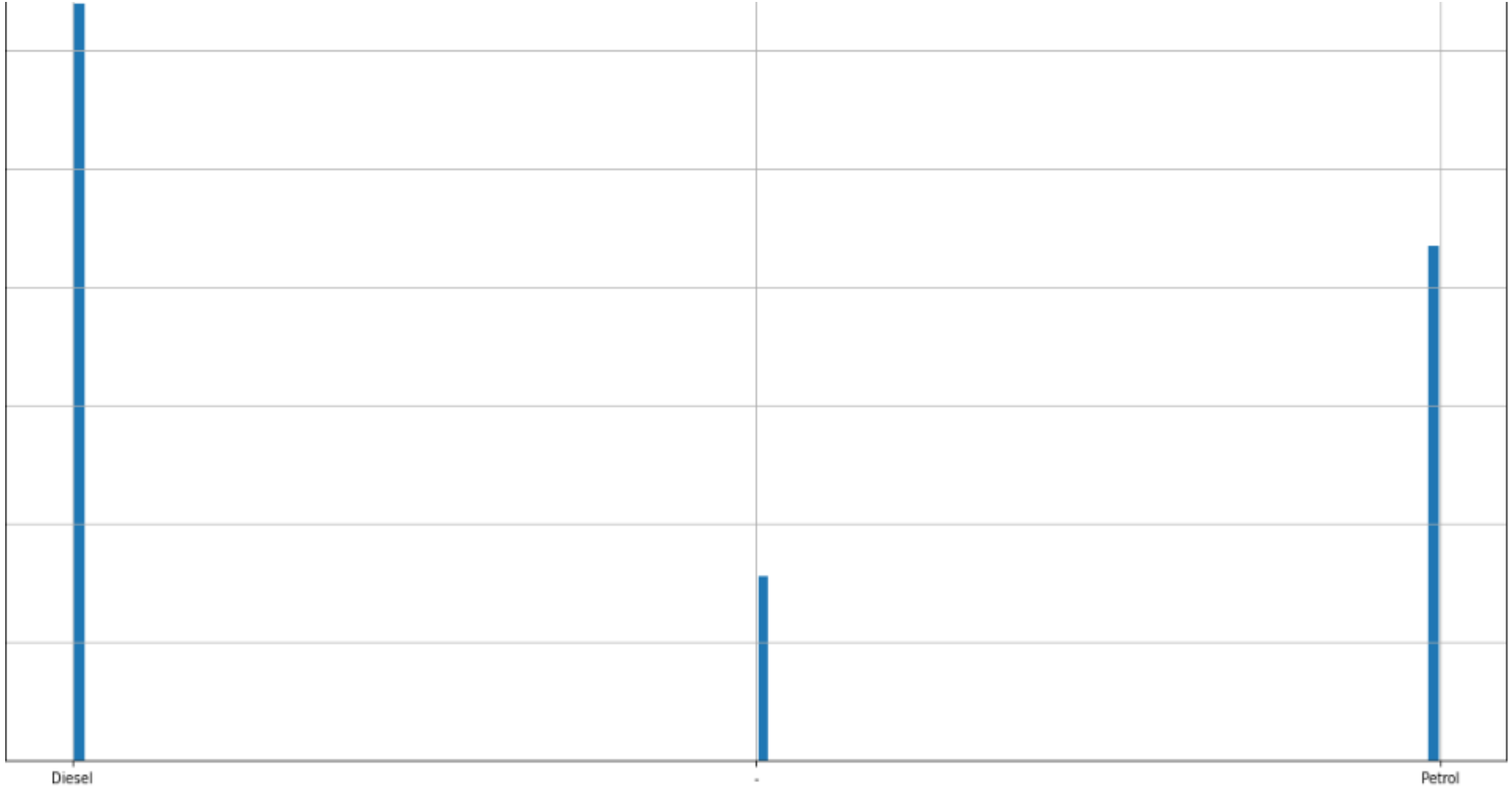
Oldest car: 2008; Newest car: 2021

Minimum price: Rs. 126,000; Maximum price of car: Rs. 29,380,000

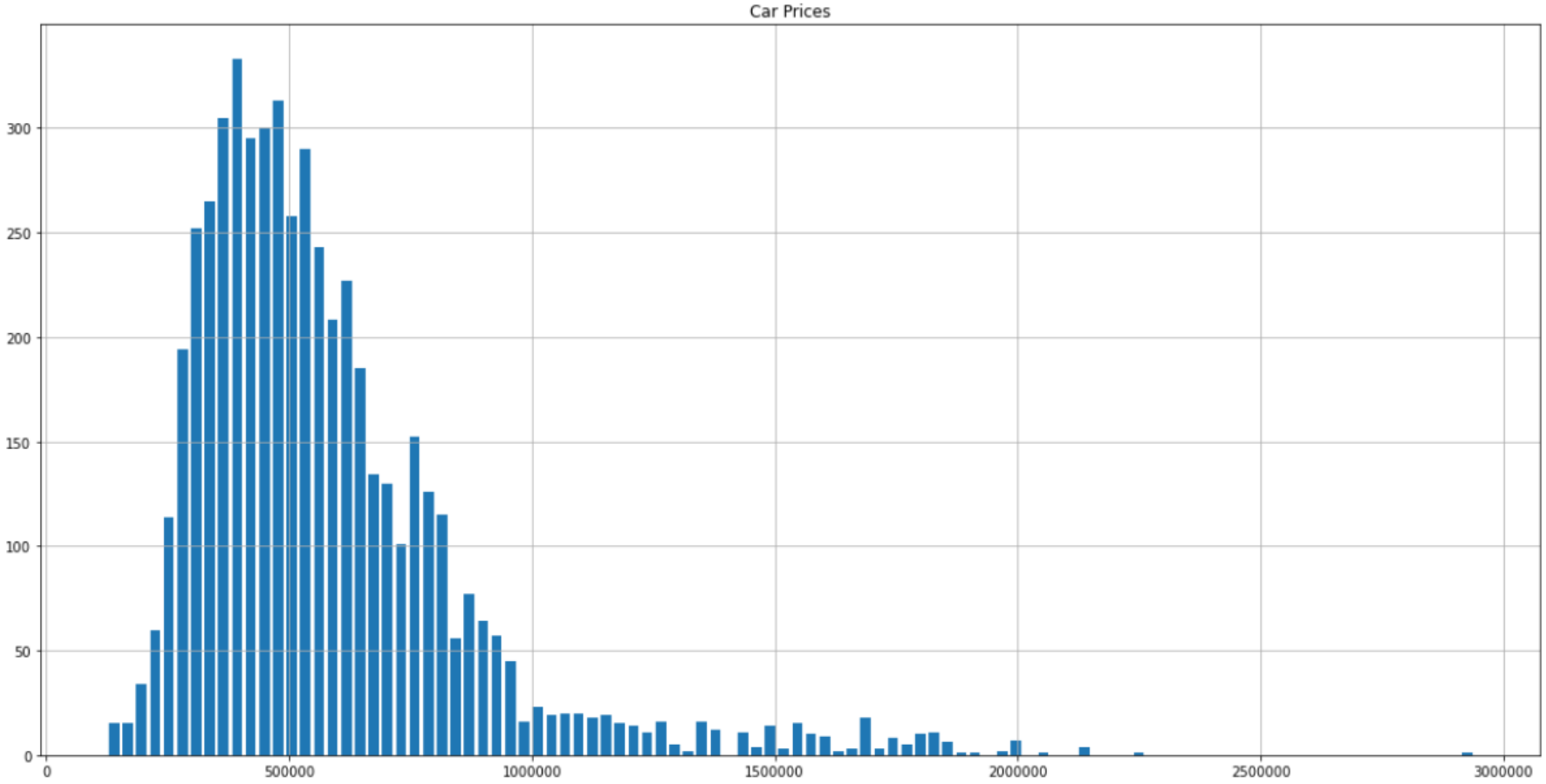
Lets dig deeper into the end points




Max mileage on this i20 car with a mileage of 1002408 and year 2010. the previous owner(s) drove a lot.

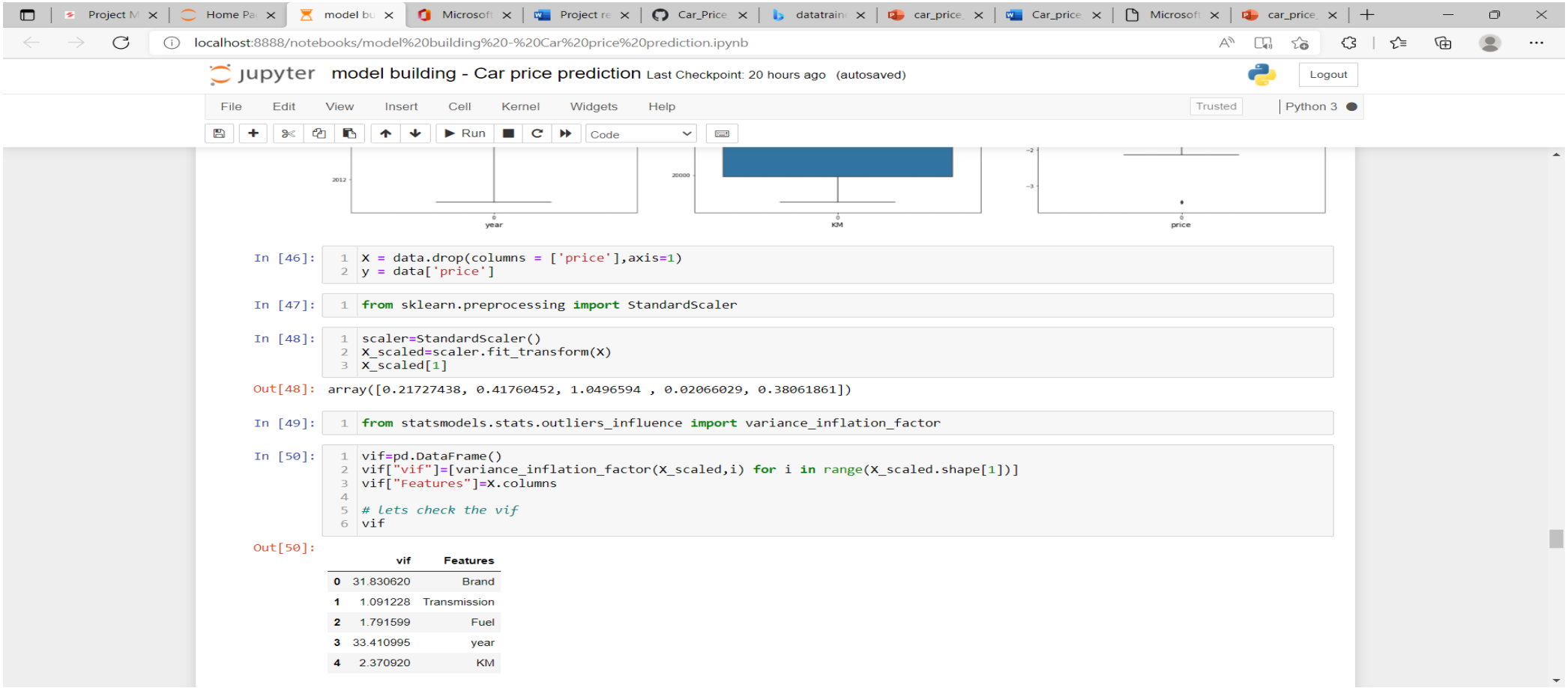

Diesel
-
Petrol

# Car prices distribution

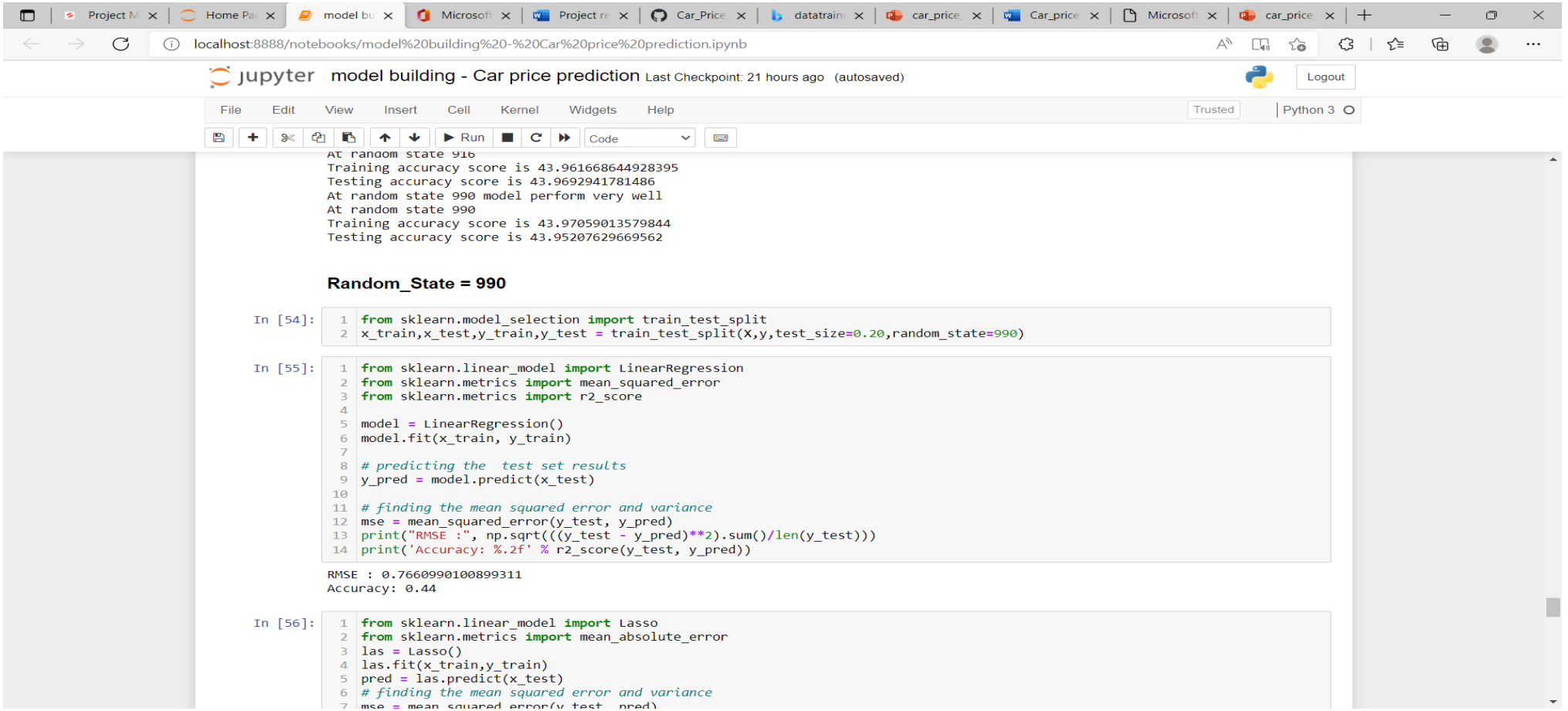



We can see we have quite some range on the prices of the cars.

# Data pre processing



# Model building



# Hyper parameter tuning

```
from sklearn.model_selection import GridSearchCV
```

```
parameters = { 'n_estimators' : [100,150],
               'min_samples_leaf' : [1,2],
              'min_samples_split': [2,3],
              'criterion': ['mse','mae']
}
```

```
GCV = GridSearchCV(RandomForestRegressor(),parameters,cv=5)
```

```
GCV.fit(X_train,y_train)
```

```
GridSearchCV(cv=5, estimator=RandomForestRegressor(),
             param_grid={'criterion': ['mse', 'mae'],
                         'min_samples_leaf': [1, 2],
                         'min_samples_split': [2, 3],
                         'n_estimators': [100, 150]})
```

```
GCV.best_params_
```

```
{'criterion': 'mse',
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'n_estimators': 150}
```

Hyper parameter tuning was performed on the best performing algorithm, which was the random forest regression.

These variables were selected, the tuning took many hours to complete.

The best parameters were used to train another accurate model.

# Training best model

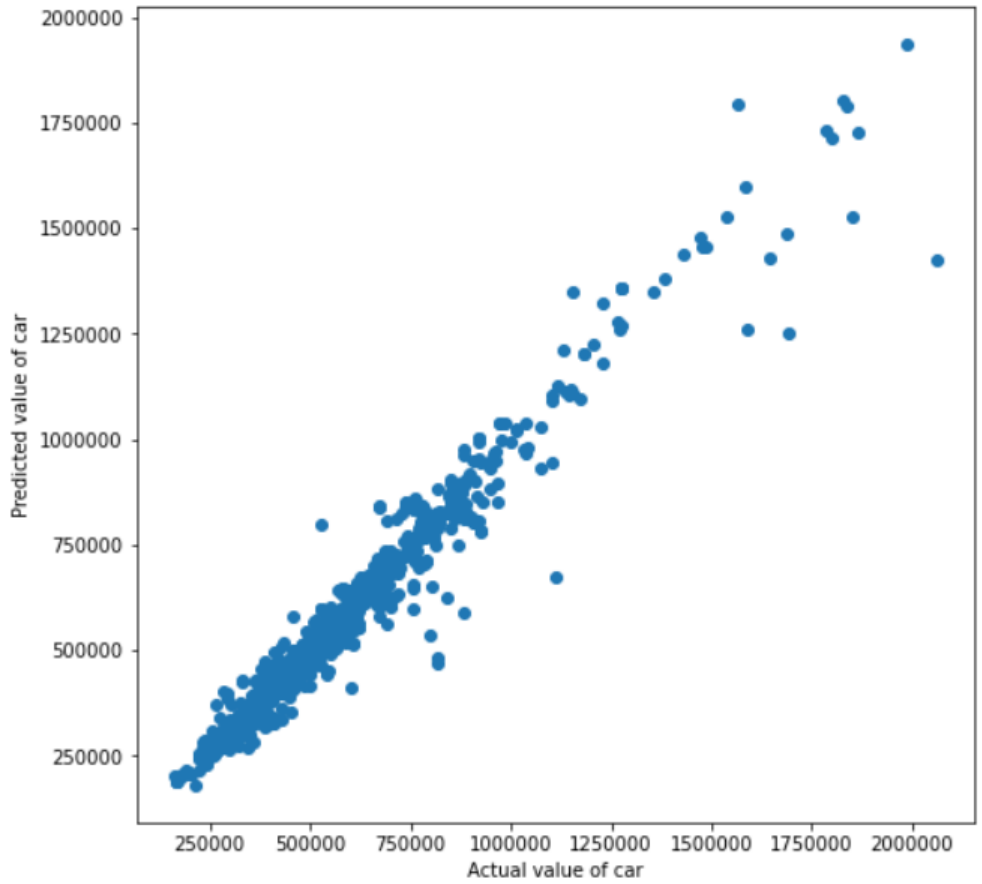
```
mod = RandomForestRegressor(min_samples_leaf= 1, min_samples_split =2, n_estimators = 150, criterion='mse')
mod.fit(X_train,y_train)
pred = mod.predict(X_test)
mod.score(X_test,y_test)
```

```
0.9630274955526736
```

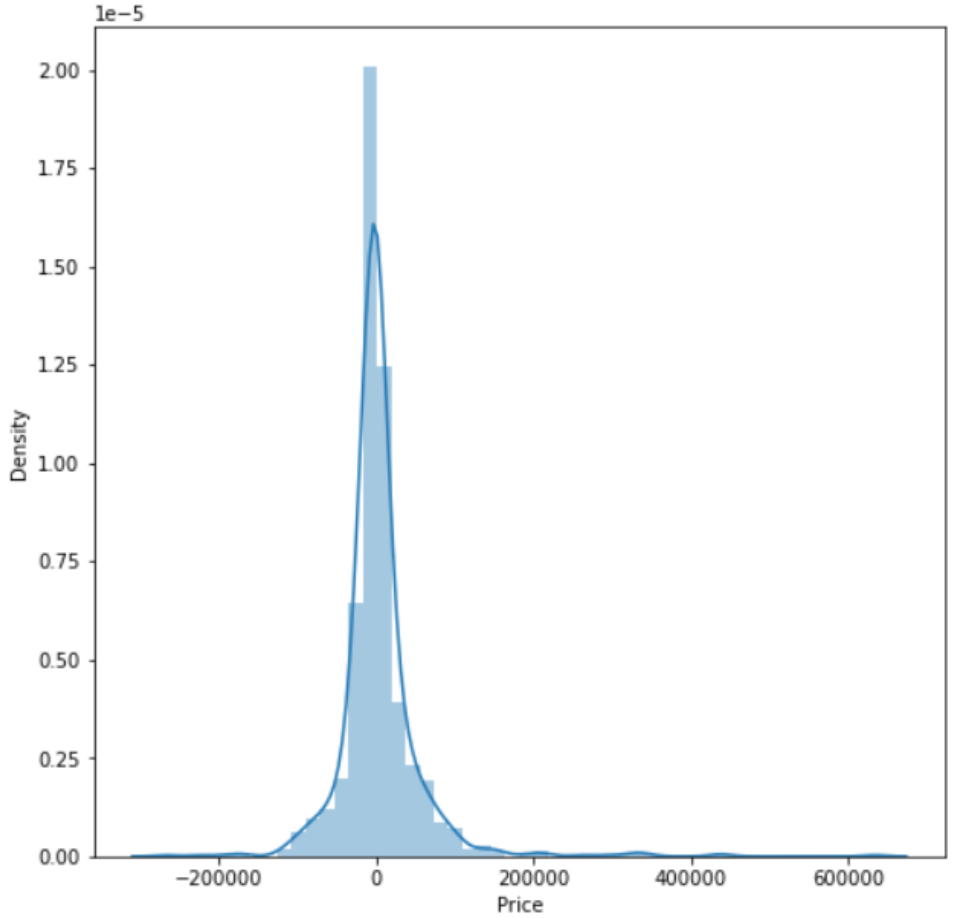
```
scr = cross_val_score(mod, X,y, cv=4)
print(scr.mean())
```

```
0.94254259734771
```

This is the best model with a cross validation, R-squared result of 0.943




Scatter plot of predicted vs actual price




Distribution plot of (predicted value – actual value)

After observing these graphs we can conclude that the model is performing very well and it is ready to be sent to the vendors.