

# LINEAR REGRESSION

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
import seaborn as sns
sns.set()
```

```
In [3]: data = pd.read_csv('1.04.+Real-life+example.csv')
data.head()
```

Out[3]:

	Brand	Price	Body	Mileage	EngineV	Engine Type	Registration	Year	Model
0	BMW	4200.0	sedan	277	2.0	Petrol	yes	1991	320
1	Mercedes-Benz	7900.0	van	427	2.9	Diesel	yes	1999	Sprinter 212
2	Mercedes-Benz	13300.0	sedan	358	5.0	Gas	yes	2003	S 500
3	Audi	23000.0	crossover	240	4.2	Petrol	yes	2007	Q7
4	Toyota	18300.0	crossover	120	2.0	Petrol	yes	2011	Rav 4

```
In [4]: data.describe(include='all')
```

Out[4]:

	Brand	Price	Body	Mileage	EngineV	Engine Type	Registration	Year	Model
count	4345	4173.000000	4345	4345.000000	4195.000000	4345	4345	4345.000000	
unique	7	NaN	6	NaN	NaN	4	2	NaN	
top	Volkswagen	NaN	sedan	NaN	NaN	Diesel	yes	NaN	
freq	936	NaN	1649	NaN	NaN	2019	3947	NaN	
mean	NaN	19418.746935	NaN	161.237284	2.790734	NaN	NaN	2006.550058	
std	NaN	25584.242620	NaN	105.705797	5.066437	NaN	NaN	6.719097	
min	NaN	600.000000	NaN	0.000000	0.600000	NaN	NaN	1969.000000	
25%	NaN	6999.000000	NaN	86.000000	1.800000	NaN	NaN	2003.000000	
50%	NaN	11500.000000	NaN	155.000000	2.200000	NaN	NaN	2008.000000	
75%	NaN	21700.000000	NaN	230.000000	3.000000	NaN	NaN	2012.000000	
max	NaN	300000.000000	NaN	980.000000	99.990000	NaN	NaN	2016.000000	

```
In [5]: dropped_data = data.drop('Model', axis=1)
dropped_data.describe(include='all')
```

Out[5]:

	Brand	Price	Body	Mileage	EngineV	Engine Type	Registration	Year
count	4345	4173.000000	4345	4345.000000	4195.000000	4345	4345	4345.000000

	Brand	Price	Body	Mileage	EngineV	Engine Type	Registration	Year
<b>unique</b>	7	NaN	6	NaN	NaN	4	2	NaN
<b>top</b>	Volkswagen	NaN	sedan	NaN	NaN	Diesel	yes	NaN
<b>freq</b>	936	NaN	1649	NaN	NaN	2019	3947	NaN
<b>mean</b>	NaN	19418.746935	NaN	161.237284	2.790734	NaN	NaN	2006.550058
<b>std</b>	NaN	25584.242620	NaN	105.705797	5.066437	NaN	NaN	6.719097
<b>min</b>	NaN	600.000000	NaN	0.000000	0.600000	NaN	NaN	1969.000000
<b>25%</b>	NaN	6999.000000	NaN	86.000000	1.800000	NaN	NaN	2003.000000
<b>50%</b>	NaN	11500.000000	NaN	155.000000	2.200000	NaN	NaN	2008.000000
<b>75%</b>	NaN	21700.000000	NaN	230.000000	3.000000	NaN	NaN	2012.000000
<b>max</b>	NaN	300000.000000	NaN	980.000000	99.990000	NaN	NaN	2016.000000

In [6]:

dropped\_data.isnull().sum()

Out[6]:

Brand0  
Price172  
Body0  
Mileage0  
EngineV150  
Engine Type0  
Registration0  
Year0  
dtype: int64

In [7]:

nomv\_data =dropped\_data.dropna(axis=0)  
nomv\_data.describe(include ='all')

Out[7]:

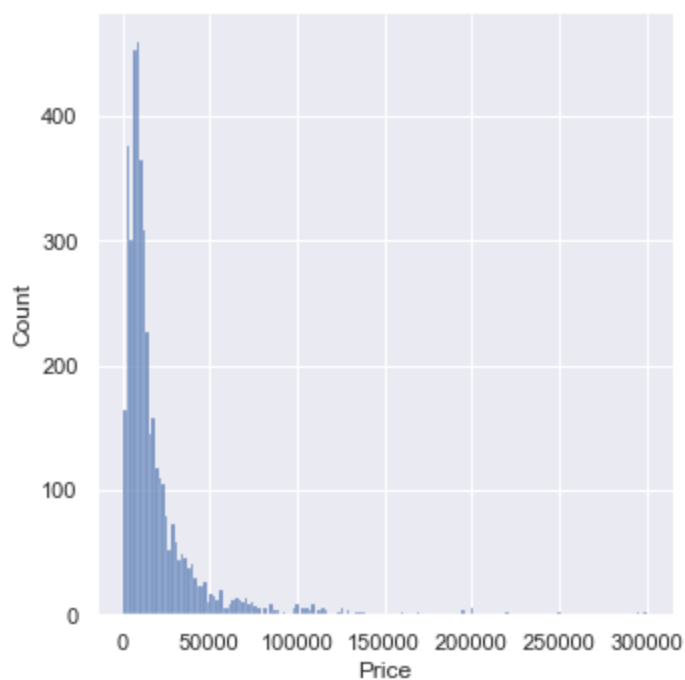
	Brand	Price	Body	Mileage	EngineV	Engine Type	Registration	Year
<b>count</b>	4025	4025.000000	4025	4025.000000	4025.000000	4025	4025	4025.000000
<b>unique</b>	7	NaN	6	NaN	NaN	4	2	NaN
<b>top</b>	Volkswagen	NaN	sedan	NaN	NaN	Diesel	yes	NaN
<b>freq</b>	880	NaN	1534	NaN	NaN	1861	3654	NaN
<b>mean</b>	NaN	19552.308065	NaN	163.572174	2.764586	NaN	NaN	2006.379627
<b>std</b>	NaN	25815.734988	NaN	103.394703	4.935941	NaN	NaN	6.695595
<b>min</b>	NaN	600.000000	NaN	0.000000	0.600000	NaN	NaN	1969.000000
<b>25%</b>	NaN	6999.000000	NaN	90.000000	1.800000	NaN	NaN	2003.000000
<b>50%</b>	NaN	11500.000000	NaN	158.000000	2.200000	NaN	NaN	2007.000000
<b>75%</b>	NaN	21900.000000	NaN	230.000000	3.000000	NaN	NaN	2012.000000
<b>max</b>	NaN	300000.000000	NaN	980.000000	99.990000	NaN	NaN	2016.000000

In [8]:

sns.displot(data['Price'])

Out[8]:

<seaborn.axisgrid.FacetGrid at 0x7fac2a773b80>



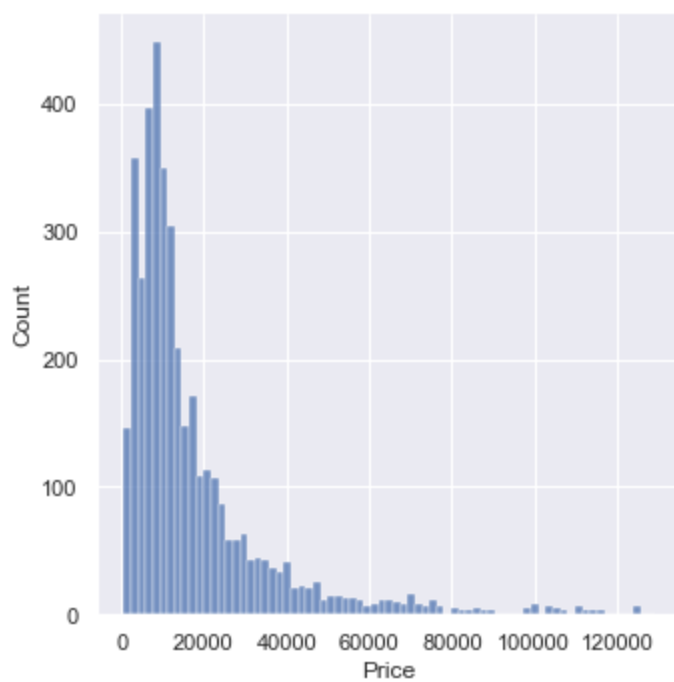
```
In [12]: q = nomv_data['Price'].quantile(0.99)
price_data = nomv_data[nomv_data['Price'] < q]
price_data.describe()
```

```
Out[12]:
```

	Price	Mileage	EngineV	Year
<b>count</b>	3984.000000	3984.000000	3984.000000	3984.000000
<b>mean</b>	17837.117460	165.116466	2.743770	2006.292922
<b>std</b>	18976.268315	102.766126	4.956057	6.672745
<b>min</b>	600.000000	0.000000	0.600000	1969.000000
<b>25%</b>	6980.000000	93.000000	1.800000	2002.750000
<b>50%</b>	11400.000000	160.000000	2.200000	2007.000000
<b>75%</b>	21000.000000	230.000000	3.000000	2011.000000
<b>max</b>	129222.000000	980.000000	99.990000	2016.000000

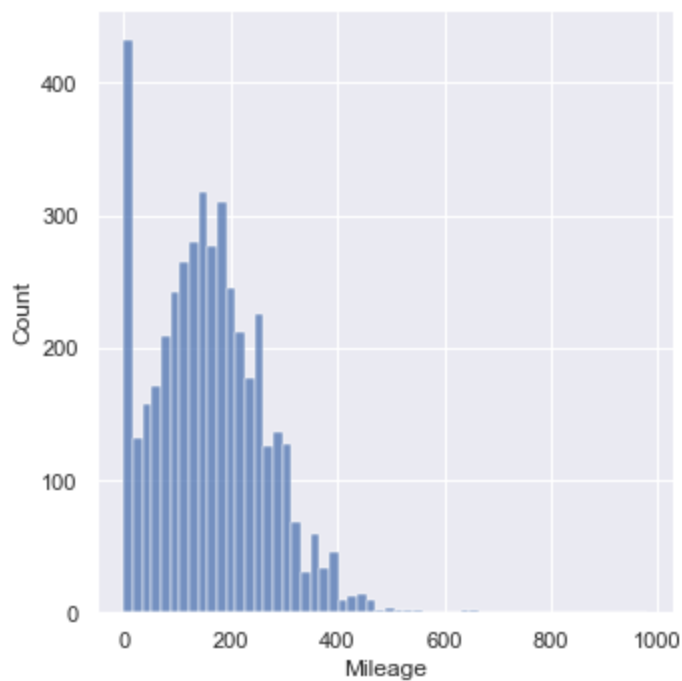
```
In [18]: sns.displot(price_data['Price'])
```

```
Out[18]: <seaborn.axisgrid.FacetGrid at 0x7fac2c2e28b0>
```



```
In [19]: sns.displot(data['Mileage'])
```

```
Out[19]: <seaborn.axisgrid.FacetGrid at 0x7fac32c3da00>
```



```
In [20]: q = price_data['Mileage'].quantile(0.99)
mileage_data = price_data[price_data['Mileage'] < q]
mileage_data.describe()
```

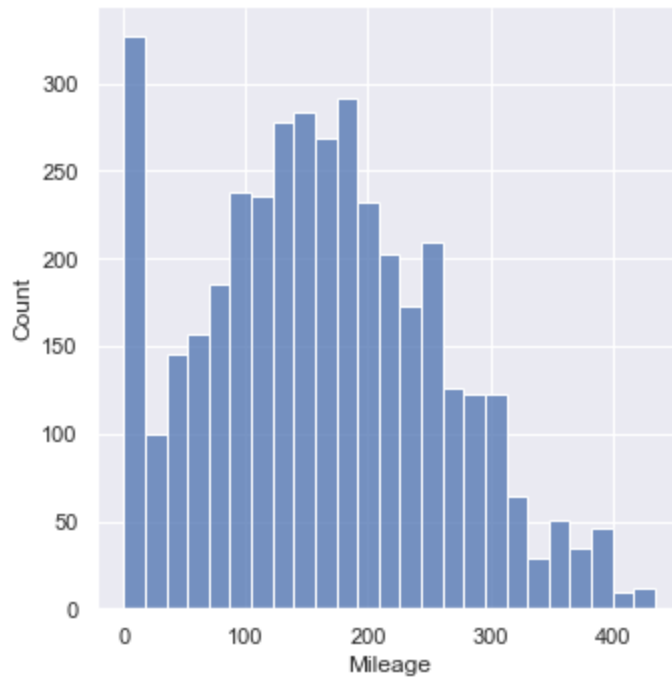
```
Out[20]:
```

	Price	Mileage	EngineV	Year
<b>count</b>	3944.000000	3944.000000	3944.000000	3944.000000
<b>mean</b>	17933.880822	161.484026	2.747612	2006.389959
<b>std</b>	19008.212025	96.027108	4.980406	6.595986
<b>min</b>	600.000000	0.000000	0.600000	1969.000000
<b>25%</b>	7000.000000	92.000000	1.800000	2003.000000

	Price	Mileage	EngineV	Year
<b>50%</b>	11500.000000	158.000000	2.200000	2007.000000
<b>75%</b>	21376.250000	230.000000	3.000000	2011.000000
<b>max</b>	129222.000000	435.000000	99.990000	2016.000000

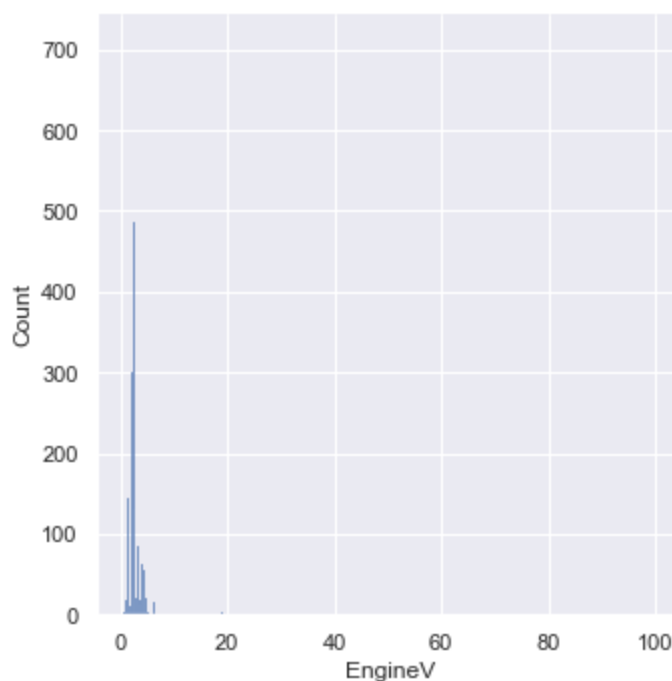
```
In [21]: sns.displot(mileage_data['Mileage'])
```

```
Out[21]: <seaborn.axisgrid.FacetGrid at 0x7fac32c64fd0>
```



```
In [25]: sns.displot(data['EngineV'])
```

```
Out[25]: <seaborn.axisgrid.FacetGrid at 0x7fac332609a0>
```



```
In [26]: engv_data = mileage_data[mileage_data['EngineV'] < 6.5]
engv_data.describe()
```

Out [26]:

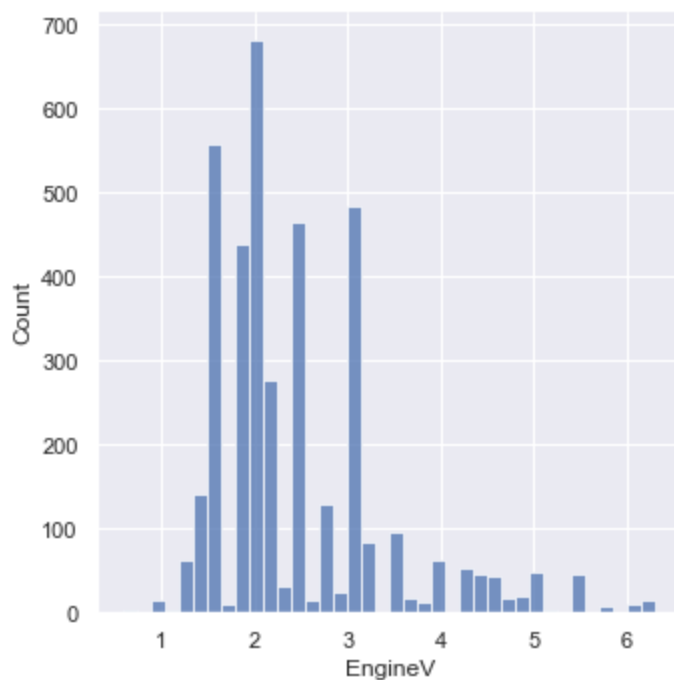
	Price	Mileage	EngineV	Year
<b>count</b>	3921.000000	3921.000000	3921.000000	3921.000000
<b>mean</b>	17988.421311	161.279776	2.443497	2006.406019
<b>std</b>	19041.658686	96.068269	0.946198	6.595871
<b>min</b>	600.000000	0.000000	0.600000	1969.000000
<b>25%</b>	7000.000000	92.000000	1.800000	2003.000000
<b>50%</b>	11500.000000	158.000000	2.200000	2008.000000
<b>75%</b>	21500.000000	229.000000	3.000000	2012.000000
<b>max</b>	129222.000000	435.000000	6.300000	2016.000000

In [27]:

```
sns.displot(engv_data['EngineV'])
```

Out [27]:

<seaborn.axisgrid.FacetGrid at 0x7fac32ed9ee0>

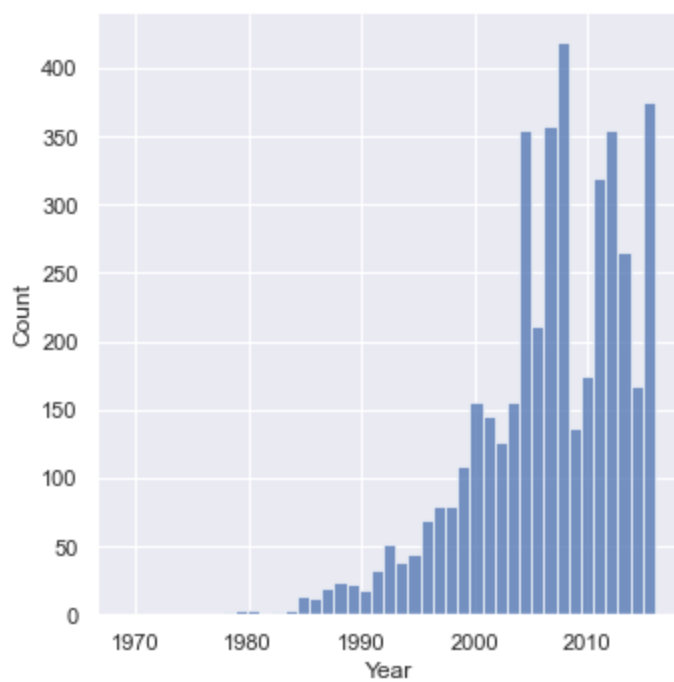


In [28]:

```
sns.displot(data['Year'])
```

Out [28]:

<seaborn.axisgrid.FacetGrid at 0x7fac31c84700>



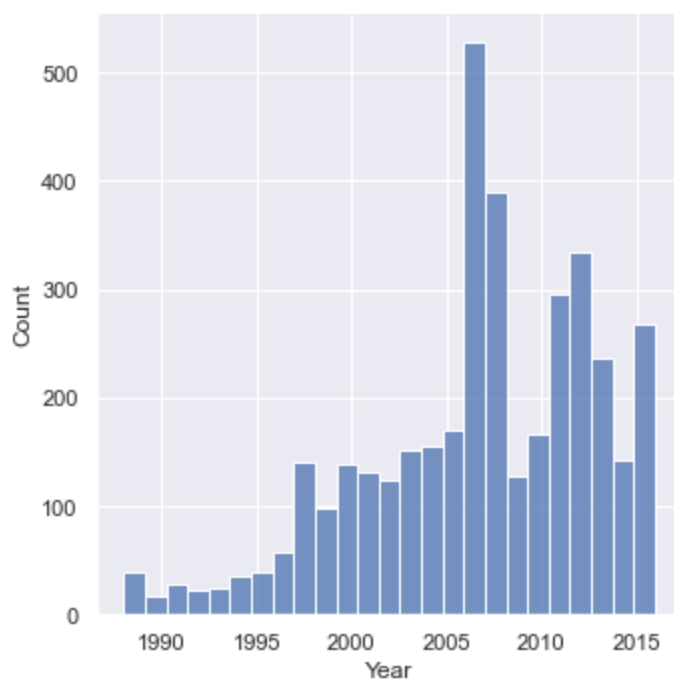
```
In [29]: q = engv_data['Year'].quantile(0.01)
year_data= engv_data[engv_data['Year']>q]
year_data.describe()
```

```
Out[29]:
```

	Price	Mileage	EngineV	Year
<b>count</b>	3867.000000	3867.000000	3867.000000	3867.000000
<b>mean</b>	18194.455679	160.542539	2.450440	2006.709853
<b>std</b>	19085.855165	95.633291	0.949366	6.103870
<b>min</b>	800.000000	0.000000	0.600000	1988.000000
<b>25%</b>	7200.000000	91.000000	1.800000	2003.000000
<b>50%</b>	11700.000000	157.000000	2.200000	2008.000000
<b>75%</b>	21700.000000	225.000000	3.000000	2012.000000
<b>max</b>	129222.000000	435.000000	6.300000	2016.000000

```
In [32]: sns.displot(year_data['Year'])
```

```
Out[32]: <seaborn.axisgrid.FacetGrid at 0x7fac33cbaa30>
```



```
In [33]: cleaned_data = year_data.reset_index(drop=True)
cleaned_data.describe(include='all')
```

```
Out[33]:
```

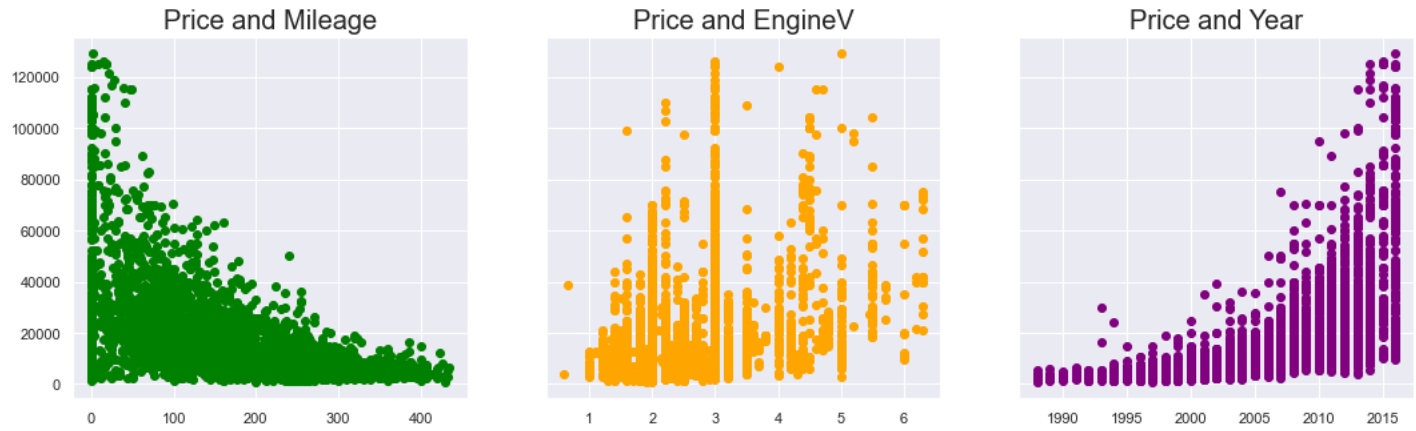
	Brand	Price	Body	Mileage	EngineV	Engine Type	Registration	Year
<b>count</b>	3867	3867.000000	3867	3867.000000	3867.000000	3867	3867	3867.000000
<b>unique</b>	7	NaN	6	NaN	NaN	4	2	NaN
<b>top</b>	Volkswagen	NaN	sedan	NaN	NaN	Diesel	yes	NaN
<b>freq</b>	848	NaN	1467	NaN	NaN	1807	3505	NaN
<b>mean</b>	NaN	18194.455679	NaN	160.542539	2.450440	NaN	NaN	2006.709853
<b>std</b>	NaN	19085.855165	NaN	95.633291	0.949366	NaN	NaN	6.103870
<b>min</b>	NaN	800.000000	NaN	0.000000	0.600000	NaN	NaN	1988.000000
<b>25%</b>	NaN	7200.000000	NaN	91.000000	1.800000	NaN	NaN	2003.000000
<b>50%</b>	NaN	11700.000000	NaN	157.000000	2.200000	NaN	NaN	2008.000000
<b>75%</b>	NaN	21700.000000	NaN	225.000000	3.000000	NaN	NaN	2012.000000
<b>max</b>	NaN	129222.000000	NaN	435.000000	6.300000	NaN	NaN	2016.000000

```
In [34]: f, (ax1, ax2, ax3) = plt.subplots(1, 3, sharey=True, figsize=(18, 5))

ax1.scatter(cleaned_data['Mileage'], cleaned_data['Price'], c='green')
ax1.set_title('Price and Mileage', fontsize=20)
ax2.scatter(cleaned_data['EngineV'], cleaned_data['Price'], c='orange')
ax2.set_title('Price and EngineV', fontsize=20)
ax3.scatter(cleaned_data['Year'], cleaned_data['Price'], c='purple')
ax3.set_title('Price and Year', fontsize=20)

plt.show()
```





```
In [35]: log_price = np.log(cleaned_data['Price'])
```

```
In [36]: cleaned_data['log_price'] = log_price
cleaned_data.describe(include='all')
```

```
Out[36]:
```

	Brand	Price	Body	Mileage	EngineV	Engine Type	Registration	Year
count	3867	3867.000000	3867	3867.000000	3867.000000	3867	3867	3867.000000
unique	7	NaN	6	NaN	NaN	4	2	NaN
top	Volkswagen	NaN	sedan	NaN	NaN	Diesel	yes	NaN
freq	848	NaN	1467	NaN	NaN	1807	3505	NaN
mean	NaN	18194.455679	NaN	160.542539	2.450440	NaN	NaN	2006.709853
std	NaN	19085.855165	NaN	95.633291	0.949366	NaN	NaN	6.103870
min	NaN	800.000000	NaN	0.000000	0.600000	NaN	NaN	1988.000000
25%	NaN	7200.000000	NaN	91.000000	1.800000	NaN	NaN	2003.000000
50%	NaN	11700.000000	NaN	157.000000	2.200000	NaN	NaN	2008.000000
75%	NaN	21700.000000	NaN	225.000000	3.000000	NaN	NaN	2012.000000
max	NaN	129222.000000	NaN	435.000000	6.300000	NaN	NaN	2016.000000

```
In [37]: cleaned_data = cleaned_data.drop(['Price'], axis = 1)
cleaned_data
```

```
Out[37]:
```

	Brand	Body	Mileage	EngineV	Engine Type	Registration	Year	log_price
0	BMW	sedan	277	2.0	Petrol	yes	1991	8.342840
1	Mercedes-Benz	van	427	2.9	Diesel	yes	1999	8.974618
2	Mercedes-Benz	sedan	358	5.0	Gas	yes	2003	9.495519
3	Audi	crossover	240	4.2	Petrol	yes	2007	10.043249
4	Toyota	crossover	120	2.0	Petrol	yes	2011	9.814656
...	...	...	...	...	...	...	...	...
3862	Volkswagen	van	163	2.5	Diesel	yes	2008	9.350102
3863	Toyota	sedan	35	1.6	Petrol	yes	2014	9.792556
3864	Mercedes-Benz	sedan	9	3.0	Diesel	yes	2014	11.736069

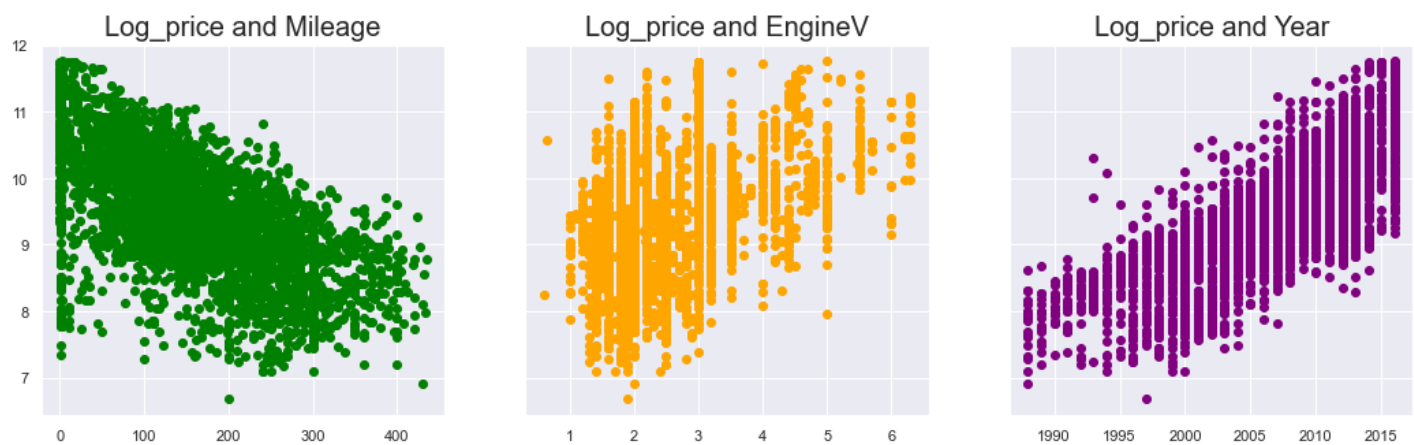
	Brand	Body	Mileage	EngineV	Engine Type	Registration	Year	log_price
3865	BMW	sedan	1	3.5	Petrol	yes	1999	8.779557
3866	Volkswagen	van	124	2.0	Diesel	yes	2013	9.510445

3867 rows x 8 columns

```
In [38]: f,(ax1,ax2,ax3) = plt.subplots(1,3, sharey=True, figsize=(18,5))

ax1.scatter(cleaned_data['Mileage'],cleaned_data['log_price'], c='green')
ax1.set_title('Log_price and Mileage', fontsize=20)
ax2.scatter(cleaned_data['EngineV'],cleaned_data['log_price'], c='orange')
ax2.set_title('Log_price and EngineV',fontsize=20)
ax3.scatter(cleaned_data['Year'],cleaned_data['log_price'],c='purple')
ax3.set_title('Log_price and Year',fontsize=20)

plt.show()
```



```
In [39]: cleaned_data.columns.values
```

```
Out[39]: array(['Brand', 'Body', 'Mileage', 'EngineV', 'Engine Type',
        'Registration', 'Year', 'log_price'], dtype=object)
```

```
In [40]: from statsmodels.stats.outliers_influence import variance_inflation_factor
```

```
In [41]: variables = cleaned_data[['Mileage','EngineV', 'Year']]
```

```
In [42]: vif = pd.DataFrame()
```

```
In [43]: vif["VIF"] = [variance_inflation_factor(variables.values, i) for i in range(variables.shape[0])]
```

```
In [160... vif['Features']= variables.columns
```

```
In [161... vif
```

```
Out[161... VIF Features
0 3.791584 Mileage
```

# VIF Features

1

7.662068

EngineV

2

10.354854

Year

In [164...

```
nomc_data= cleaned_data.drop(['Year'], axis=1)
nomc_data
```

Out[164...

	Brand	Body	Mileage	EngineV	Engine Type	Registration	log_price
0	BMW	sedan	277	2.0	Petrol	yes	8.342840
1	Mercedes-Benz	van	427	2.9	Diesel	yes	8.974618
2	Mercedes-Benz	sedan	358	5.0	Gas	yes	9.495519
3	Audi	crossover	240	4.2	Petrol	yes	10.043249
4	Toyota	crossover	120	2.0	Petrol	yes	9.814656
...	...	...	...	...	...	...	...
3862	Volkswagen	van	163	2.5	Diesel	yes	9.350102
3863	Toyota	sedan	35	1.6	Petrol	yes	9.792556
3864	Mercedes-Benz	sedan	9	3.0	Diesel	yes	11.736069
3865	BMW	sedan	1	3.5	Petrol	yes	8.779557
3866	Volkswagen	van	124	2.0	Diesel	yes	9.510445

3867 rows × 7 columns

In [166...

```
nomc_data.describe(include='all')
```

Out[166...

	Brand	Body	Mileage	EngineV	Engine Type	Registration	log_price
count	3867	3867	3867.000000	3867.000000	3867	3867	3867.000000
unique	7	6	NaN	NaN	4	2	NaN
top	Volkswagen	sedan	NaN	NaN	Diesel	yes	NaN
freq	848	1467	NaN	NaN	1807	3505	NaN
mean	NaN	NaN	160.542539	2.450440	NaN	NaN	9.415703
std	NaN	NaN	95.633291	0.949366	NaN	NaN	0.876616
min	NaN	NaN	0.000000	0.600000	NaN	NaN	6.684612
25%	NaN	NaN	91.000000	1.800000	NaN	NaN	8.881836
50%	NaN	NaN	157.000000	2.200000	NaN	NaN	9.367344
75%	NaN	NaN	225.000000	3.000000	NaN	NaN	9.985068
max	NaN	NaN	435.000000	6.300000	NaN	NaN	11.769287

In [168...

```
dummies_data = pd.get_dummies(nomc_data, drop_first=True)
dummies_data
```

Out[168...

	Mileage	EngineV	log_price	Brand_BMW	Brand_Mercedes-Benz	Brand_Mitsubishi	Brand_Renault	Brand
--	---------	---------	-----------	-----------	---------------------	------------------	---------------	-------

	Mileage	EngineV	log_price	Brand_BMW	Brand_Mercedes-Benz	Brand_Mitsubishi	Brand_Renault	Brand
0	277	2.0	8.342840	1	0	0	0	
1	427	2.9	8.974618	0	1	0	0	
2	358	5.0	9.495519	0	1	0	0	
3	240	4.2	10.043249	0	0	0	0	
4	120	2.0	9.814656	0	0	0	0	
...	...	...	...	...	...	...	...	...
3862	163	2.5	9.350102	0	0	0	0	
3863	35	1.6	9.792556	0	0	0	0	
3864	9	3.0	11.736069	0	1	0	0	
3865	1	3.5	8.779557	1	0	0	0	
3866	124	2.0	9.510445	0	0	0	0	

3867 rows × 18 columns

In [169...

```
dummies_data.columns.values
```

Out[169...

```
array(['Mileage', 'EngineV', 'log_price', 'Brand_BMW',
      'Brand_Mercedes-Benz', 'Brand_Mitsubishi', 'Brand_Renault',
      'Brand_Toyota', 'Brand_Volkswagen', 'Body_hatch', 'Body_other',
      'Body_sedan', 'Body_vagon', 'Body_van', 'Engine Type_Gas',
      'Engine Type_Other', 'Engine Type_Petrol', 'Registration_yes'],
      dtype=object)
```

In [170...

```
cols = ['log_price', 'Mileage', 'EngineV', 'Brand_BMW',
        'Brand_Mercedes-Benz', 'Brand_Mitsubishi', 'Brand_Renault',
        'Brand_Toyota', 'Brand_Volkswagen', 'Body_hatch', 'Body_other',
        'Body_sedan', 'Body_vagon', 'Body_van', 'Engine Type_Gas',
        'Engine Type_Other', 'Engine Type_Petrol', 'Registration_yes']
```

In [171...

```
preprocessed_data = dummies_data[cols]
preprocessed_data.head()
```

Out[171...

	log_price	Mileage	EngineV	Brand_BMW	Brand_Mercedes-Benz	Brand_Mitsubishi	Brand_Renault	Brand_To
0	8.342840	277	2.0	1	0	0	0	
1	8.974618	427	2.9	0	1	0	0	
2	9.495519	358	5.0	0	1	0	0	
3	10.043249	240	4.2	0	0	0	0	
4	9.814656	120	2.0	0	0	0	0	

In [173...

```
targets = preprocessed_data['log_price']
features= preprocessed_data.drop(['log_price'], axis=1)
```

In [174...

```
from sklearn.preprocessing import StandardScaler
```

```
In [175... scaler = StandardScaler()
```

```
In [176... scaler.fit(features)
```

```
Out[176... StandardScaler()
```

```
In [177... features_scaled = scaler.transform(features)
```

```
In [178... from sklearn.model_selection import train_test_split
```

```
In [179... x_train,x_test,y_train,y_test = train_test_split(features_scaled,targets,test_size =0.2,ra
```

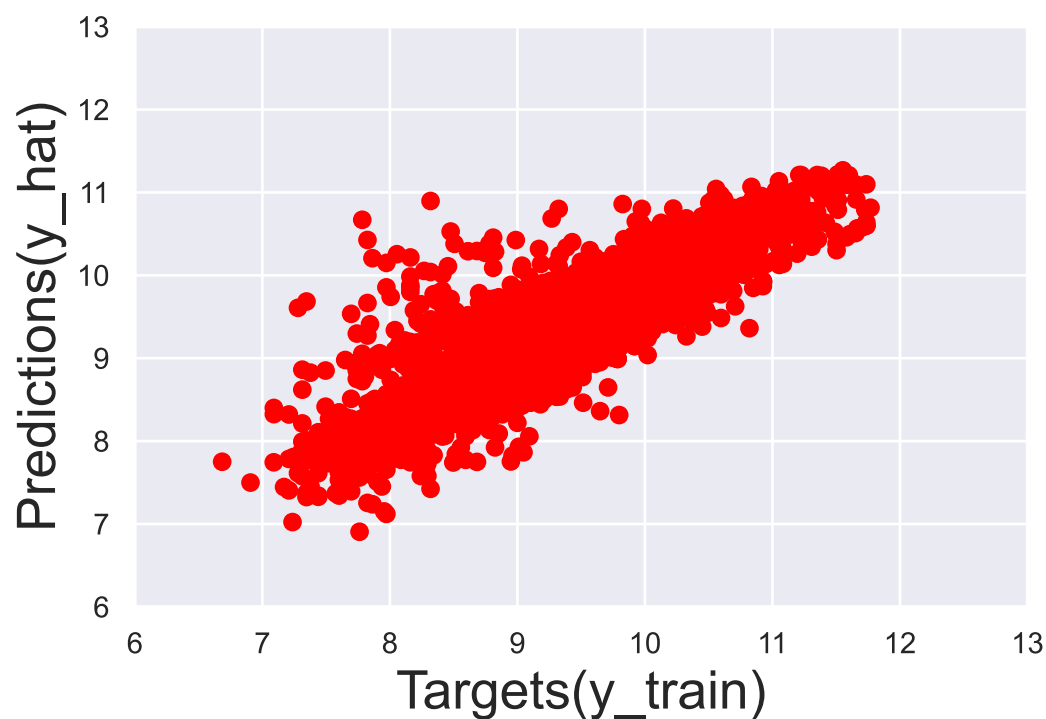
```
In [180... reg =LinearRegression()
```

```
In [181... reg.fit(x_train,y_train)
```

```
Out[181... LinearRegression()
```

```
In [182... y_hat = reg.predict(x_train)
```

```
In [183... plt.scatter(y_train,y_hat, c= 'red')  
plt.xlabel('Targets(y_train)',size =20)  
plt.ylabel('Predictions(y_hat)', size =20)  
plt.xlim(6,13)  
plt.ylim(6,13)  
plt.show()
```



```
In [184... sns.distplot(y_train-y_hat)
```

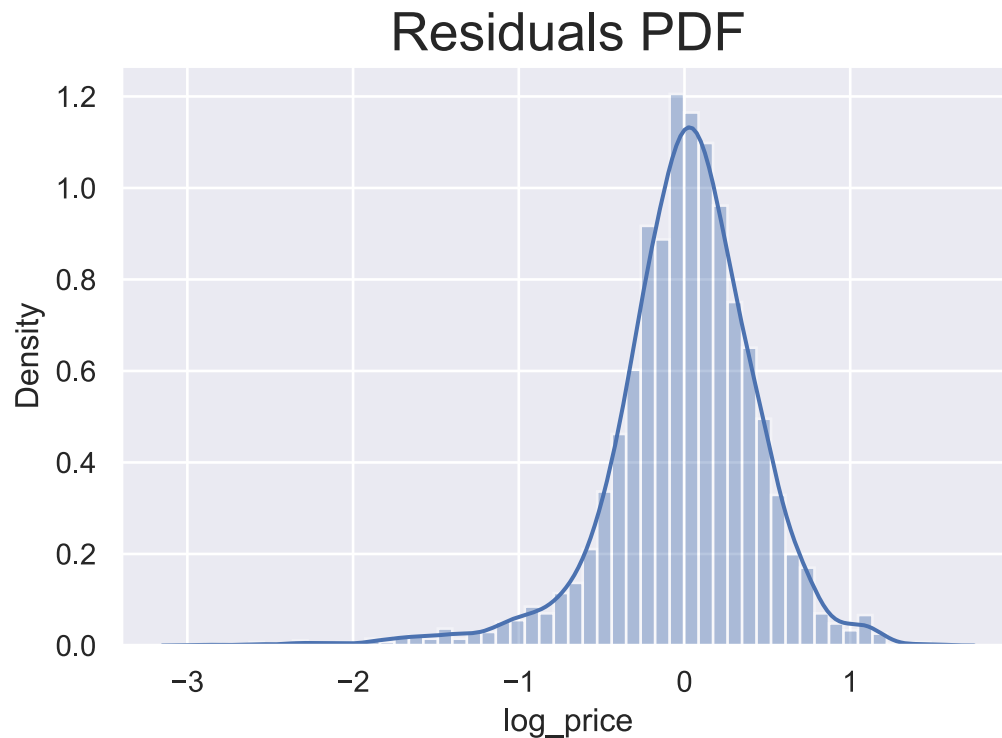
```
plt.title('Residuals PDF', size=20)
```

/opt/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

```
Text(0.5, 1.0, 'Residuals PDF')
```

Out[184...



In [185...

```
reg.score(x_train,y_train)
```

Out[185...

```
0.744996578792662
```

In [186...

```
def adj_r2(x_train,y_train):  
    r2 = reg.score(x_train,y_train)  
    n = x_train.shape[0]  
    p = x_train.shape[1]  
    adjusted_r2 = 1-(1-r2)*(n-1)/(n-p-1)  
    return adjusted_r2
```

In [187...

```
adj_r2(x_train,y_train)
```

Out[187...

```
0.7435868037811093
```

In [188...

```
reg.intercept_
```

Out[188...

```
9.415239458021299
```

In [189...

```
reg.coef_
```

Out[189...

```
array([-0.44871341,  0.20903483,  0.0142496 ,  0.01288174, -0.14055166,  
       -0.17990912, -0.06054988, -0.08992433, -0.1454692 , -0.10144383,       ...])
```

```
-0.20062984, -0.12988747, -0.16859669, -0.12149035, -0.03336798,  
-0.14690868, 0.32047333])
```

In [190...

```
reg.summary=pd.DataFrame(features.columns.values, columns=['Features'])  
reg.summary['Weights'] = reg.coef_  
reg.summary
```

Out[190...

	Features	Weights
0	Mileage	-0.448713
1	EngineV	0.209035
2	Brand_BMW	0.014250
3	Brand_Mercedes-Benz	0.012882
4	Brand_Mitsubishi	-0.140552
5	Brand_Renault	-0.179909
6	Brand_Toyota	-0.060550
7	Brand_Volkswagen	-0.089924
8	Body_hatch	-0.145469
9	Body_other	-0.101444
10	Body_sedan	-0.200630
11	Body_vagon	-0.129887
12	Body_van	-0.168597
13	Engine Type_Gas	-0.121490
14	Engine Type_Other	-0.033368
15	Engine Type_Petrol	-0.146909
16	Registration_yes	0.320473

In [191...

```
cleaned_data['Brand'].unique()
```

Out[191...

```
array(['BMW', 'Mercedes-Benz', 'Audi', 'Toyota', 'Renault', 'Volkswagen',  
      'Mitsubishi'], dtype=object)
```

In [192...

```
cleaned_data['Registration'].unique()
```

Out[192...

```
array(['yes', 'no'], dtype=object)
```

In [193...

```
cleaned_data['Body'].unique()
```

Out[193...

```
array(['sedan', 'van', 'crossover', 'vagon', 'other', 'hatch'],  
      dtype=object)
```

In [194...

```
cleaned_data['Engine Type'].unique()
```

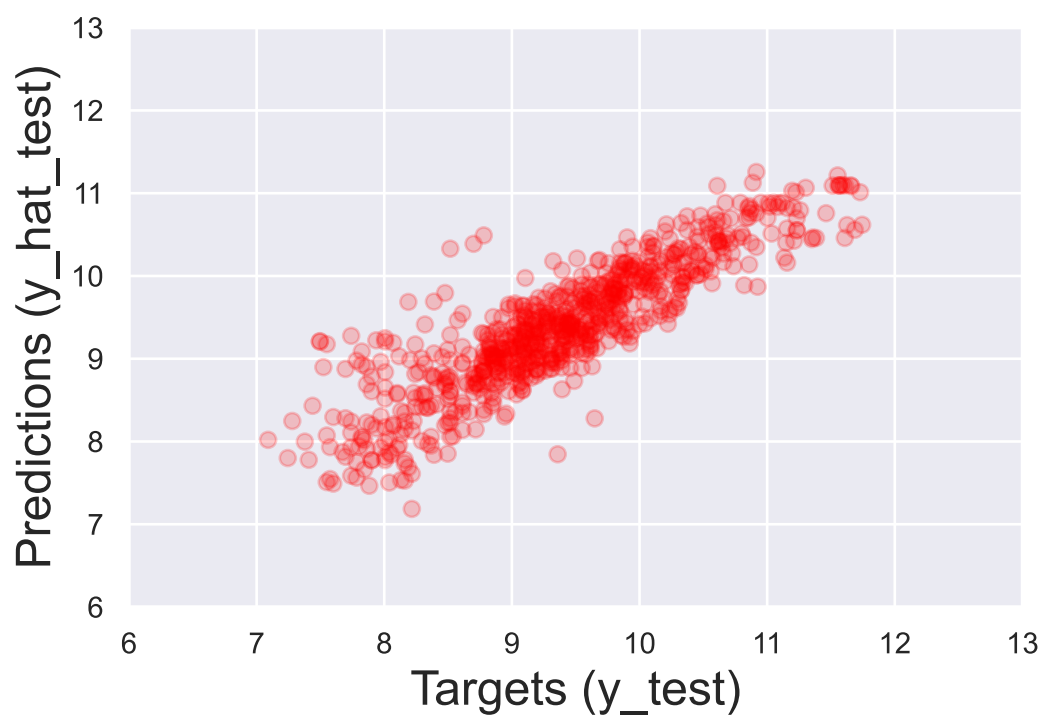
Out[194...

```
array(['Petrol', 'Diesel', 'Gas', 'Other'], dtype=object)
```

In [195...

```
y_hat_test = reg.predict(x_test)
```

```
In [201... plt.scatter(y_test, y_hat_test, c= 'red', alpha=0.2)
plt.xlabel('Targets (y_test)',size=18)
plt.ylabel('Predictions (y_hat_test)',size=18)
plt.xlim(6,13)
plt.ylim(6,13)
plt.show()
```



```
In [202... df_pf = pd.DataFrame(np.exp(y_hat_test), columns=['Prediction'])
df_pf.head()
```

Out [202...

	Prediction
0	10685.501696
1	3499.255242
2	7553.285218
3	7463.963017
4	11353.490075

```
In [203... df_pf['Target'] = np.exp(y_test)
df_pf
```

Out [203...

	Prediction	Target
0	10685.501696	NaN
1	3499.255242	7900.0
2	7553.285218	NaN
3	7463.963017	NaN
4	11353.490075	NaN
...	...	...
769	29651.726363	6950.0



	Prediction	Target
<b>770</b>	10732.071179	NaN
<b>771</b>	13922.446953	NaN
<b>772</b>	27487.751303	NaN
<b>773</b>	13491.163043	NaN

774 rows × 2 columns

```
In [209... y_test = y_test.reset_index(drop=True)

y_test.head()
```

```
Out[209... 0    7.740664
1    7.937375
2    7.824046
3    8.764053
4    9.121509
Name: log_price, dtype: float64
```

```
In [205... df_pf['Residual'] = df_pf['Target'] - df_pf['Prediction']
```

```
In [207... df_pf['Difference%'] = np.absolute(df_pf['Residual']/df_pf['Target']*100)
df_pf
```

```
Out[207...

```

	Prediction	Target	Residual	Difference%
<b>0</b>	10685.501696	NaN	NaN	NaN
<b>1</b>	3499.255242	7900.0	4400.744758	55.705630
<b>2</b>	7553.285218	NaN	NaN	NaN
<b>3</b>	7463.963017	NaN	NaN	NaN
<b>4</b>	11353.490075	NaN	NaN	NaN
...	...	...	...	...
<b>769</b>	29651.726363	6950.0	-22701.726363	326.643545
<b>770</b>	10732.071179	NaN	NaN	NaN
<b>771</b>	13922.446953	NaN	NaN	NaN
<b>772</b>	27487.751303	NaN	NaN	NaN
<b>773</b>	13491.163043	NaN	NaN	NaN

774 rows × 4 columns

```
In [210... df_pf.describe()
```

```
Out[210...

```

	Prediction	Target	Residual	Difference%
<b>count</b>	774.000000	147.000000	147.000000	147.000000
<b>mean</b>	15946.760167	14463.840340	-2245.724613	160.027708
<b>std</b>	13133.197604	14242.311606	20036.962794	285.224728

	Prediction	Target	Residual	Difference%
<b>min</b>	1320.562768	1400.000000	-62998.368303	0.287114
<b>25%</b>	7413.644234	6600.000000	-12990.588876	36.783373
<b>50%</b>	11568.168859	10700.000000	-1826.259439	68.902496
<b>75%</b>	20162.408805	17825.000000	6300.488157	173.709107
<b>max</b>	77403.055224	109999.000000	99435.606951	2571.361972

In [211...

```
pd.options.display.max_rows = 999
pd.set_option('display.float_format', lambda x: '%.2f' % x)
df_pf.sort_values(by=['Difference%'])
```

Out [211...

	Prediction	Target	Residual	Difference%
<b>109</b>	12663.54	12700.00	36.46	0.29
<b>606</b>	20680.40	20500.00	-180.40	0.88
<b>433</b>	6977.90	6900.00	-77.90	1.13
<b>8</b>	11581.54	11950.00	368.46	3.08
<b>330</b>	11336.71	10800.00	-536.71	4.97
<b>440</b>	8638.63	8200.00	-438.63	5.35
<b>84</b>	4364.08	4000.00	-364.08	9.10
<b>31</b>	12858.08	14500.00	1641.92	11.32
<b>313</b>	10014.71	11600.00	1585.29	13.67
<b>122</b>	19335.34	22700.00	3364.66	14.82
<b>714</b>	23118.07	19700.00	-3418.07	17.35
<b>745</b>	6421.18	7777.00	1355.82	17.43
<b>544</b>	12354.92	15000.00	2645.08	17.63
<b>371</b>	9661.78	12000.00	2338.22	19.49
<b>747</b>	8453.28	10500.00	2046.72	19.49
<b>117</b>	7178.20	8990.00	1811.80	20.15
<b>128</b>	7727.53	9800.00	2072.47	21.15
<b>687</b>	11004.05	8999.00	-2005.05	22.28
<b>463</b>	35093.27	28500.00	-6593.27	23.13
<b>404</b>	8648.80	6999.00	-1649.80	23.57
<b>151</b>	10889.39	8800.00	-2089.39	23.74
<b>466</b>	25975.12	34500.00	8524.88	24.71
<b>503</b>	6392.70	8500.00	2107.30	24.79
<b>305</b>	10505.98	13999.00	3493.02	24.95
<b>703</b>	5597.06	7500.00	1902.94	25.37
<b>376</b>	5391.77	4300.00	-1091.77	25.39
<b>401</b>	9070.23	12300.00	3229.77	26.26
<b>96</b>	7976.26	6150.00	-1826.26	29.70

	Prediction	Target	Residual	Difference%
705	12976.15	18700.00	5723.85	30.61
190	13377.99	19999.00	6621.01	33.11
237	25037.87	18800.00	-6237.87	33.18
576	15954.72	11900.00	-4054.72	34.07
498	7950.27	12300.00	4349.73	35.36
73	18577.88	13708.50	-4869.38	35.52
138	11095.57	17300.00	6204.43	35.86
90	9143.17	6700.00	-2443.17	36.47
334	12414.72	19600.00	7185.28	36.66
149	8351.34	6100.00	-2251.34	36.91
98	6746.48	4900.00	-1846.48	37.68
554	8340.94	13500.00	5159.06	38.22
752	8591.66	13999.00	5407.34	38.63
514	15015.22	10800.00	-4215.22	39.03
682	5454.34	8999.00	3544.66	39.39
325	9057.77	15000.00	5942.23	39.61
728	5792.10	9800.00	4007.90	40.90
199	2784.46	4999.00	2214.54	44.30
86	8071.70	14500.00	6428.30	44.33
758	9380.46	16999.00	7618.54	44.82
340	14815.83	10200.00	-4615.83	45.25
578	3205.38	2200.00	-1005.38	45.70
653	10332.54	19400.00	9067.46	46.74
241	18432.63	34900.00	16467.37	47.18
749	6082.85	4100.00	-1982.85	48.36
5	21289.80	14200.00	-7089.80	49.93
11	5175.77	10500.00	5324.23	50.71
248	22011.47	44999.00	22987.53	51.08
169	8522.39	17650.00	9127.61	51.71
586	9348.90	19500.00	10151.10	52.06
352	5238.05	11500.00	6261.95	54.45
40	18672.68	41000.00	22327.32	54.46
171	8392.42	18500.00	10107.58	54.64
1	3499.26	7900.00	4400.74	55.71
212	16133.86	10200.00	-5933.86	58.18
159	12904.68	8000.00	-4904.68	61.31
465	6537.82	16999.00	10461.18	61.54
581	3246.94	8650.00	5403.06	62.46

	Prediction	Target	Residual	Difference%
393	13600.39	36900.00	23299.61	63.14
357	11018.79	29900.00	18881.21	63.15
233	6110.94	16900.00	10789.06	63.84
438	16257.03	9900.00	-6357.03	64.21
167	13212.52	38888.00	25675.48	66.02
734	23336.06	13999.00	-9337.06	66.70
665	17858.02	10700.00	-7158.02	66.90
720	2860.97	9200.00	6339.03	68.90
253	18128.32	61000.00	42871.68	70.28
532	10019.90	34900.00	24880.10	71.29
378	2749.08	9600.00	6850.92	71.36
51	2543.57	8900.00	6356.43	71.42
271	3948.74	13900.00	9951.26	71.59
491	6503.62	25749.75	19246.13	74.74
566	11881.05	51500.00	39618.95	76.93
142	6839.57	31600.00	24760.43	78.36
425	3940.49	18988.13	15047.64	79.25
211	6848.88	33200.00	26351.12	79.37
722	30409.09	16950.00	-13459.09	79.40
475	5570.46	27800.00	22229.54	79.96
68	6717.17	3700.00	-3017.17	81.55
710	3795.06	20859.15	17064.09	81.81
452	39882.67	21900.00	-17982.67	82.11
696	2470.61	15500.00	13029.39	84.06
69	2440.90	15700.00	13259.10	84.45
126	13459.92	87777.00	74317.08	84.67
459	2779.04	18800.00	16020.96	85.22
389	2810.71	22400.00	19589.29	87.45
592	28154.84	14900.00	-13254.84	88.96
674	2381.01	24000.00	21618.99	90.08
658	10563.39	109999.00	99435.61	90.40
59	8195.34	4300.00	-3895.34	90.59
362	27293.13	13893.75	-13399.38	96.44
507	18282.75	8420.00	-9862.75	117.13
546	6746.48	3100.00	-3646.48	117.63
307	24974.57	11100.00	-13874.57	125.00
257	41503.90	18000.00	-23503.90	130.58
19	16178.14	6800.00	-9378.14	137.91

	Prediction	Target	Residual	Difference%
391	25302.43	10500.00	-14802.43	140.98
345	28951.20	11700.00	-17251.20	147.45
311	51287.19	20500.00	-30787.19	150.18
274	9113.05	3600.00	-5513.05	153.14
762	12201.29	4700.00	-7501.29	159.60
681	5195.69	1900.00	-3295.69	173.46
328	15067.85	5500.00	-9567.85	173.96
148	25040.42	8900.00	-16140.42	181.35
603	14513.46	5000.00	-9513.46	190.27
765	11034.66	3750.00	-7284.66	194.26
387	44173.72	15000.00	-29173.72	194.49
275	22547.55	7500.00	-15047.55	200.63
590	14512.61	4799.00	-9713.61	202.41
501	60493.51	18999.00	-41494.51	218.40
474	10932.17	3400.00	-7532.17	221.53
531	47047.70	14600.00	-32447.70	222.24
454	20972.54	6500.00	-14472.54	222.65
560	33172.20	10000.00	-23172.20	231.72
43	21611.83	6500.00	-15111.83	232.49
296	21690.99	6500.00	-15190.99	233.71
410	16407.29	4900.00	-11507.29	234.84
95	47013.28	13500.00	-33513.28	248.25
266	65742.57	18800.00	-46942.57	249.69
769	29651.73	6950.00	-22701.73	326.64
18	37600.15	7500.00	-30100.15	401.34
620	25752.19	5100.00	-20652.19	404.94
175	34863.53	6800.00	-28063.53	412.70
162	9954.42	1900.00	-8054.42	423.92
652	17195.57	3200.00	-13995.57	437.36
639	30628.28	5500.00	-25128.28	456.88
120	50728.89	8900.00	-41828.89	469.99
736	53312.81	8500.00	-44812.81	527.21
180	19701.70	3000.00	-16701.70	556.72
460	50954.68	7650.00	-43304.68	566.07
381	33499.44	4999.00	-28500.44	570.12
299	20183.30	2999.00	-17184.30	573.00
16	23721.58	3500.00	-20221.58	577.76
476	15700.53	2300.00	-13400.53	582.63

	Prediction	Target	Residual	Difference%
150	9800.43	1400.00	-8400.43	600.03
420	14576.33	1850.00	-12726.33	687.91
595	25576.58	2700.00	-22876.58	847.28
422	53294.61	3149.25	-50145.36	1592.30
408	65448.37	2450.00	-62998.37	2571.36
0	10685.50	NaN	NaN	NaN
2	7553.29	NaN	NaN	NaN
3	7463.96	NaN	NaN	NaN
4	11353.49	NaN	NaN	NaN
6	20159.19	NaN	NaN	NaN
7	20349.62	NaN	NaN	NaN
9	33614.62	NaN	NaN	NaN
10	7241.07	NaN	NaN	NaN
12	5484.02	NaN	NaN	NaN
13	13292.71	NaN	NaN	NaN
14	8248.67	NaN	NaN	NaN
15	10621.84	NaN	NaN	NaN
17	11770.64	NaN	NaN	NaN
20	11876.82	NaN	NaN	NaN
21	31557.80	NaN	NaN	NaN
22	6102.36	NaN	NaN	NaN
23	13111.91	NaN	NaN	NaN
24	23650.15	NaN	NaN	NaN
25	45272.25	NaN	NaN	NaN
26	2178.94	NaN	NaN	NaN
27	2555.02	NaN	NaN	NaN
28	35991.51	NaN	NaN	NaN
29	26062.23	NaN	NaN	NaN
30	15559.43	NaN	NaN	NaN
32	12459.85	NaN	NaN	NaN
33	22941.80	NaN	NaN	NaN
34	19854.79	NaN	NaN	NaN
35	11337.51	NaN	NaN	NaN
36	10499.60	NaN	NaN	NaN
37	26478.15	NaN	NaN	NaN
38	9919.19	NaN	NaN	NaN
39	16724.45	NaN	NaN	NaN
41	19845.36	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
42	23286.27	NaN	NaN	NaN
44	7241.07	NaN	NaN	NaN
45	8512.03	NaN	NaN	NaN
46	3542.72	NaN	NaN	NaN
47	26425.14	NaN	NaN	NaN
48	77403.06	NaN	NaN	NaN
49	4973.17	NaN	NaN	NaN
50	2479.43	NaN	NaN	NaN
52	10716.84	NaN	NaN	NaN
53	20453.89	NaN	NaN	NaN
54	18575.43	NaN	NaN	NaN
55	12674.57	NaN	NaN	NaN
56	15380.04	NaN	NaN	NaN
57	21942.05	NaN	NaN	NaN
58	35336.33	NaN	NaN	NaN
60	12469.21	NaN	NaN	NaN
61	7396.87	NaN	NaN	NaN
62	10848.85	NaN	NaN	NaN
63	5351.86	NaN	NaN	NaN
64	9611.80	NaN	NaN	NaN
65	19847.25	NaN	NaN	NaN
66	3316.44	NaN	NaN	NaN
67	7621.25	NaN	NaN	NaN
70	50206.83	NaN	NaN	NaN
71	15627.03	NaN	NaN	NaN
72	23624.69	NaN	NaN	NaN
74	1746.19	NaN	NaN	NaN
75	4209.17	NaN	NaN	NaN
76	5144.02	NaN	NaN	NaN
77	5930.73	NaN	NaN	NaN
78	30810.25	NaN	NaN	NaN
79	18893.76	NaN	NaN	NaN
80	6819.44	NaN	NaN	NaN
81	12891.96	NaN	NaN	NaN
82	7320.42	NaN	NaN	NaN
83	6013.05	NaN	NaN	NaN
85	19001.29	NaN	NaN	NaN
87	2315.71	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
88	10807.78	NaN	NaN	NaN
89	11696.25	NaN	NaN	NaN
91	13421.16	NaN	NaN	NaN
92	7116.68	NaN	NaN	NaN
93	18317.20	NaN	NaN	NaN
94	7724.17	NaN	NaN	NaN
97	3648.96	NaN	NaN	NaN
99	6437.21	NaN	NaN	NaN
100	32886.86	NaN	NaN	NaN
101	5958.63	NaN	NaN	NaN
102	28625.56	NaN	NaN	NaN
103	4488.43	NaN	NaN	NaN
104	5521.28	NaN	NaN	NaN
105	8882.65	NaN	NaN	NaN
106	16547.03	NaN	NaN	NaN
107	7131.92	NaN	NaN	NaN
108	7300.05	NaN	NaN	NaN
110	25614.14	NaN	NaN	NaN
111	3152.62	NaN	NaN	NaN
112	22835.48	NaN	NaN	NaN
113	6006.27	NaN	NaN	NaN
114	27716.14	NaN	NaN	NaN
115	6981.34	NaN	NaN	NaN
116	15819.50	NaN	NaN	NaN
118	14943.75	NaN	NaN	NaN
119	10448.50	NaN	NaN	NaN
121	41075.73	NaN	NaN	NaN
123	5643.45	NaN	NaN	NaN
124	15691.29	NaN	NaN	NaN
125	13733.33	NaN	NaN	NaN
127	12311.47	NaN	NaN	NaN
129	11580.54	NaN	NaN	NaN
130	17489.92	NaN	NaN	NaN
131	23967.81	NaN	NaN	NaN
132	11200.20	NaN	NaN	NaN
133	9279.28	NaN	NaN	NaN
134	12096.18	NaN	NaN	NaN
135	44407.14	NaN	NaN	NaN



	Prediction	Target	Residual	Difference%
136	12006.66	NaN	NaN	NaN
137	66032.76	NaN	NaN	NaN
139	7712.65	NaN	NaN	NaN
140	10358.58	NaN	NaN	NaN
141	1815.28	NaN	NaN	NaN
143	8768.29	NaN	NaN	NaN
144	11969.66	NaN	NaN	NaN
145	4307.68	NaN	NaN	NaN
146	8298.32	NaN	NaN	NaN
147	7772.57	NaN	NaN	NaN
152	44680.77	NaN	NaN	NaN
153	3666.31	NaN	NaN	NaN
154	11443.85	NaN	NaN	NaN
155	3086.38	NaN	NaN	NaN
156	9192.06	NaN	NaN	NaN
157	7491.01	NaN	NaN	NaN
158	15623.39	NaN	NaN	NaN
160	32954.19	NaN	NaN	NaN
161	13967.55	NaN	NaN	NaN
163	23972.56	NaN	NaN	NaN
164	18403.35	NaN	NaN	NaN
165	9596.94	NaN	NaN	NaN
166	15292.10	NaN	NaN	NaN
168	11420.95	NaN	NaN	NaN
170	19741.65	NaN	NaN	NaN
172	11156.38	NaN	NaN	NaN
173	36516.35	NaN	NaN	NaN
174	7745.02	NaN	NaN	NaN
176	6732.97	NaN	NaN	NaN
177	8916.19	NaN	NaN	NaN
178	17316.59	NaN	NaN	NaN
179	14933.54	NaN	NaN	NaN
181	17597.10	NaN	NaN	NaN
182	10123.04	NaN	NaN	NaN
183	47014.52	NaN	NaN	NaN
184	29479.97	NaN	NaN	NaN
185	6130.96	NaN	NaN	NaN
186	19071.43	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
187	28256.79	NaN	NaN	NaN
188	10159.18	NaN	NaN	NaN
189	10786.38	NaN	NaN	NaN
191	15956.95	NaN	NaN	NaN
192	18285.07	NaN	NaN	NaN
193	14832.32	NaN	NaN	NaN
194	9310.69	NaN	NaN	NaN
195	20588.57	NaN	NaN	NaN
196	5948.04	NaN	NaN	NaN
197	2126.49	NaN	NaN	NaN
198	17983.10	NaN	NaN	NaN
200	11098.18	NaN	NaN	NaN
201	10094.58	NaN	NaN	NaN
202	5040.46	NaN	NaN	NaN
203	65613.31	NaN	NaN	NaN
204	7664.32	NaN	NaN	NaN
205	13041.39	NaN	NaN	NaN
206	19920.63	NaN	NaN	NaN
207	15010.51	NaN	NaN	NaN
208	5716.14	NaN	NaN	NaN
209	8242.62	NaN	NaN	NaN
210	31216.83	NaN	NaN	NaN
213	15872.91	NaN	NaN	NaN
214	2393.34	NaN	NaN	NaN
215	5284.06	NaN	NaN	NaN
216	12974.89	NaN	NaN	NaN
217	8661.95	NaN	NaN	NaN
218	11490.44	NaN	NaN	NaN
219	34376.16	NaN	NaN	NaN
220	13348.91	NaN	NaN	NaN
221	34863.53	NaN	NaN	NaN
222	20662.46	NaN	NaN	NaN
223	6199.59	NaN	NaN	NaN
224	3939.91	NaN	NaN	NaN
225	6916.13	NaN	NaN	NaN
226	11191.05	NaN	NaN	NaN
227	17251.60	NaN	NaN	NaN
228	15682.82	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
229	7719.69	NaN	NaN	NaN
230	9198.08	NaN	NaN	NaN
231	9312.21	NaN	NaN	NaN
232	1320.56	NaN	NaN	NaN
234	12731.26	NaN	NaN	NaN
235	2765.14	NaN	NaN	NaN
236	9024.05	NaN	NaN	NaN
238	26736.21	NaN	NaN	NaN
239	10581.62	NaN	NaN	NaN
240	10419.20	NaN	NaN	NaN
242	7597.39	NaN	NaN	NaN
243	32925.60	NaN	NaN	NaN
244	7110.32	NaN	NaN	NaN
245	15457.91	NaN	NaN	NaN
246	41711.18	NaN	NaN	NaN
247	13159.16	NaN	NaN	NaN
249	7391.67	NaN	NaN	NaN
250	5041.20	NaN	NaN	NaN
251	6978.91	NaN	NaN	NaN
252	9721.50	NaN	NaN	NaN
254	6413.26	NaN	NaN	NaN
255	24796.09	NaN	NaN	NaN
256	1825.44	NaN	NaN	NaN
258	13680.48	NaN	NaN	NaN
259	6067.79	NaN	NaN	NaN
260	6858.89	NaN	NaN	NaN
261	53312.81	NaN	NaN	NaN
262	12414.41	NaN	NaN	NaN
263	9089.35	NaN	NaN	NaN
264	31059.18	NaN	NaN	NaN
265	20163.48	NaN	NaN	NaN
267	40753.58	NaN	NaN	NaN
268	6893.62	NaN	NaN	NaN
269	4151.02	NaN	NaN	NaN
270	8263.95	NaN	NaN	NaN
272	9749.53	NaN	NaN	NaN
273	12318.39	NaN	NaN	NaN
276	5002.39	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
277	12900.31	NaN	NaN	NaN
278	53063.22	NaN	NaN	NaN
279	3154.48	NaN	NaN	NaN
280	12327.10	NaN	NaN	NaN
281	11638.89	NaN	NaN	NaN
282	12261.19	NaN	NaN	NaN
283	7630.64	NaN	NaN	NaN
284	40765.33	NaN	NaN	NaN
285	34761.74	NaN	NaN	NaN
286	19659.97	NaN	NaN	NaN
287	1972.28	NaN	NaN	NaN
288	6635.73	NaN	NaN	NaN
289	7178.20	NaN	NaN	NaN
290	1894.40	NaN	NaN	NaN
291	20433.23	NaN	NaN	NaN
292	8962.29	NaN	NaN	NaN
293	8431.89	NaN	NaN	NaN
294	13471.84	NaN	NaN	NaN
295	5350.91	NaN	NaN	NaN
297	22261.50	NaN	NaN	NaN
298	17937.36	NaN	NaN	NaN
300	20236.85	NaN	NaN	NaN
301	9782.47	NaN	NaN	NaN
302	4738.89	NaN	NaN	NaN
303	23496.36	NaN	NaN	NaN
304	6546.81	NaN	NaN	NaN
306	3164.66	NaN	NaN	NaN
308	8967.74	NaN	NaN	NaN
309	6510.80	NaN	NaN	NaN
310	9205.10	NaN	NaN	NaN
312	14792.29	NaN	NaN	NaN
314	5811.74	NaN	NaN	NaN
315	4590.49	NaN	NaN	NaN
316	65887.51	NaN	NaN	NaN
317	12247.90	NaN	NaN	NaN
318	39744.80	NaN	NaN	NaN
319	10612.51	NaN	NaN	NaN
320	10350.25	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
321	23630.85	NaN	NaN	NaN
322	65210.35	NaN	NaN	NaN
323	7098.53	NaN	NaN	NaN
324	8316.92	NaN	NaN	NaN
326	9356.43	NaN	NaN	NaN
327	10716.00	NaN	NaN	NaN
329	7327.18	NaN	NaN	NaN
331	11002.01	NaN	NaN	NaN
332	14591.35	NaN	NaN	NaN
333	12452.22	NaN	NaN	NaN
335	13004.34	NaN	NaN	NaN
336	8250.47	NaN	NaN	NaN
337	5374.74	NaN	NaN	NaN
338	38260.36	NaN	NaN	NaN
339	11099.09	NaN	NaN	NaN
341	12897.92	NaN	NaN	NaN
342	14572.70	NaN	NaN	NaN
343	60603.06	NaN	NaN	NaN
344	11937.31	NaN	NaN	NaN
346	32945.80	NaN	NaN	NaN
347	26441.03	NaN	NaN	NaN
348	12201.29	NaN	NaN	NaN
349	2020.25	NaN	NaN	NaN
350	20246.04	NaN	NaN	NaN
351	25951.31	NaN	NaN	NaN
353	9753.44	NaN	NaN	NaN
354	16717.08	NaN	NaN	NaN
355	22454.74	NaN	NaN	NaN
356	10133.68	NaN	NaN	NaN
358	63921.33	NaN	NaN	NaN
359	4524.77	NaN	NaN	NaN
360	4770.52	NaN	NaN	NaN
361	8688.92	NaN	NaN	NaN
363	44193.38	NaN	NaN	NaN
364	21681.00	NaN	NaN	NaN
365	10547.82	NaN	NaN	NaN
366	13420.15	NaN	NaN	NaN
367	42703.68	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
368	8415.81	NaN	NaN	NaN
369	3084.69	NaN	NaN	NaN
370	10835.95	NaN	NaN	NaN
372	8110.06	NaN	NaN	NaN
373	10821.19	NaN	NaN	NaN
374	24427.14	NaN	NaN	NaN
375	65742.57	NaN	NaN	NaN
377	9909.62	NaN	NaN	NaN
379	17654.84	NaN	NaN	NaN
380	3473.79	NaN	NaN	NaN
382	7918.89	NaN	NaN	NaN
383	13757.58	NaN	NaN	NaN
384	4296.77	NaN	NaN	NaN
385	13235.55	NaN	NaN	NaN
386	7267.36	NaN	NaN	NaN
388	11474.22	NaN	NaN	NaN
390	24271.90	NaN	NaN	NaN
392	8139.48	NaN	NaN	NaN
394	9076.42	NaN	NaN	NaN
395	21303.57	NaN	NaN	NaN
396	16165.79	NaN	NaN	NaN
397	7973.87	NaN	NaN	NaN
398	8707.13	NaN	NaN	NaN
399	14410.15	NaN	NaN	NaN
400	25986.31	NaN	NaN	NaN
402	8850.21	NaN	NaN	NaN
403	2839.03	NaN	NaN	NaN
405	6425.00	NaN	NaN	NaN
406	24874.86	NaN	NaN	NaN
407	21097.66	NaN	NaN	NaN
409	8121.22	NaN	NaN	NaN
411	13409.47	NaN	NaN	NaN
412	7609.28	NaN	NaN	NaN
413	12234.68	NaN	NaN	NaN
414	14363.76	NaN	NaN	NaN
415	8117.44	NaN	NaN	NaN
416	22367.52	NaN	NaN	NaN
417	3795.06	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
418	11378.43	NaN	NaN	NaN
419	17591.09	NaN	NaN	NaN
421	8033.61	NaN	NaN	NaN
423	74232.55	NaN	NaN	NaN
424	50074.56	NaN	NaN	NaN
426	13542.79	NaN	NaN	NaN
427	5462.52	NaN	NaN	NaN
428	2389.14	NaN	NaN	NaN
429	67952.07	NaN	NaN	NaN
430	14197.26	NaN	NaN	NaN
431	16253.48	NaN	NaN	NaN
432	2975.40	NaN	NaN	NaN
434	8843.22	NaN	NaN	NaN
435	53063.22	NaN	NaN	NaN
436	40893.84	NaN	NaN	NaN
437	18866.50	NaN	NaN	NaN
439	14926.69	NaN	NaN	NaN
441	11619.16	NaN	NaN	NaN
442	25951.31	NaN	NaN	NaN
443	3735.04	NaN	NaN	NaN
444	38334.67	NaN	NaN	NaN
445	7712.65	NaN	NaN	NaN
446	9363.27	NaN	NaN	NaN
447	34863.53	NaN	NaN	NaN
448	13502.35	NaN	NaN	NaN
449	9495.62	NaN	NaN	NaN
450	14218.94	NaN	NaN	NaN
451	35956.50	NaN	NaN	NaN
453	8381.60	NaN	NaN	NaN
455	10946.44	NaN	NaN	NaN
456	19624.73	NaN	NaN	NaN
457	16821.54	NaN	NaN	NaN
458	14136.01	NaN	NaN	NaN
461	31656.01	NaN	NaN	NaN
462	43236.92	NaN	NaN	NaN
464	4266.27	NaN	NaN	NaN
467	22262.80	NaN	NaN	NaN
468	12558.90	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
469	65825.25	NaN	NaN	NaN
470	7045.72	NaN	NaN	NaN
471	4209.17	NaN	NaN	NaN
472	20367.25	NaN	NaN	NaN
473	1792.36	NaN	NaN	NaN
477	15374.08	NaN	NaN	NaN
478	36764.38	NaN	NaN	NaN
479	36883.77	NaN	NaN	NaN
480	31369.37	NaN	NaN	NaN
481	6599.62	NaN	NaN	NaN
482	12027.56	NaN	NaN	NaN
483	28937.50	NaN	NaN	NaN
484	13334.19	NaN	NaN	NaN
485	9664.46	NaN	NaN	NaN
486	13884.26	NaN	NaN	NaN
487	10824.89	NaN	NaN	NaN
488	4043.74	NaN	NaN	NaN
489	28910.96	NaN	NaN	NaN
490	8902.52	NaN	NaN	NaN
492	7161.67	NaN	NaN	NaN
493	5484.97	NaN	NaN	NaN
494	33392.40	NaN	NaN	NaN
495	3747.21	NaN	NaN	NaN
496	17540.72	NaN	NaN	NaN
497	50099.92	NaN	NaN	NaN
499	12270.01	NaN	NaN	NaN
500	8973.16	NaN	NaN	NaN
502	9984.87	NaN	NaN	NaN
504	23272.83	NaN	NaN	NaN
505	10105.70	NaN	NaN	NaN
506	3590.97	NaN	NaN	NaN
508	16197.42	NaN	NaN	NaN
509	5839.08	NaN	NaN	NaN
510	1873.03	NaN	NaN	NaN
511	17282.06	NaN	NaN	NaN
512	15144.28	NaN	NaN	NaN
513	5346.97	NaN	NaN	NaN
515	32121.20	NaN	NaN	NaN



	Prediction	Target	Residual	Difference%
516	11858.48	NaN	NaN	NaN
517	9424.34	NaN	NaN	NaN
518	21297.65	NaN	NaN	NaN
519	8667.12	NaN	NaN	NaN
520	16773.62	NaN	NaN	NaN
521	27740.81	NaN	NaN	NaN
522	17011.43	NaN	NaN	NaN
523	10809.48	NaN	NaN	NaN
524	5863.41	NaN	NaN	NaN
525	2373.66	NaN	NaN	NaN
526	29049.27	NaN	NaN	NaN
527	9829.53	NaN	NaN	NaN
528	18555.09	NaN	NaN	NaN
529	53312.81	NaN	NaN	NaN
530	17714.62	NaN	NaN	NaN
533	24642.50	NaN	NaN	NaN
534	16939.86	NaN	NaN	NaN
535	34678.95	NaN	NaN	NaN
536	6746.43	NaN	NaN	NaN
537	9139.04	NaN	NaN	NaN
538	17591.09	NaN	NaN	NaN
539	6781.86	NaN	NaN	NaN
540	4666.05	NaN	NaN	NaN
541	9369.66	NaN	NaN	NaN
542	5947.51	NaN	NaN	NaN
543	9489.56	NaN	NaN	NaN
545	6826.78	NaN	NaN	NaN
547	9175.06	NaN	NaN	NaN
548	46465.02	NaN	NaN	NaN
549	3816.33	NaN	NaN	NaN
550	14915.98	NaN	NaN	NaN
551	11555.80	NaN	NaN	NaN
552	5983.44	NaN	NaN	NaN
553	16392.22	NaN	NaN	NaN
555	6610.04	NaN	NaN	NaN
556	18231.44	NaN	NaN	NaN
557	12523.73	NaN	NaN	NaN
558	7098.53	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
559	11377.84	NaN	NaN	NaN
561	6429.03	NaN	NaN	NaN
562	9593.52	NaN	NaN	NaN
563	2285.24	NaN	NaN	NaN
564	25682.58	NaN	NaN	NaN
565	7324.63	NaN	NaN	NaN
567	7279.97	NaN	NaN	NaN
568	7940.44	NaN	NaN	NaN
569	1926.60	NaN	NaN	NaN
570	23163.87	NaN	NaN	NaN
571	26331.41	NaN	NaN	NaN
572	10597.72	NaN	NaN	NaN
573	33707.27	NaN	NaN	NaN
574	12583.52	NaN	NaN	NaN
575	1858.34	NaN	NaN	NaN
577	14486.38	NaN	NaN	NaN
579	3411.53	NaN	NaN	NaN
580	6205.48	NaN	NaN	NaN
582	7724.17	NaN	NaN	NaN
583	3425.23	NaN	NaN	NaN
584	11432.63	NaN	NaN	NaN
585	39074.36	NaN	NaN	NaN
587	15259.81	NaN	NaN	NaN
588	12255.45	NaN	NaN	NaN
589	3173.48	NaN	NaN	NaN
591	10115.13	NaN	NaN	NaN
593	17908.00	NaN	NaN	NaN
594	3001.37	NaN	NaN	NaN
596	25685.67	NaN	NaN	NaN
597	7797.18	NaN	NaN	NaN
598	8767.54	NaN	NaN	NaN
599	4590.16	NaN	NaN	NaN
600	21816.43	NaN	NaN	NaN
601	35371.16	NaN	NaN	NaN
602	16427.12	NaN	NaN	NaN
604	31373.47	NaN	NaN	NaN
605	8940.39	NaN	NaN	NaN
607	28498.91	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
608	29127.72	NaN	NaN	NaN
609	3040.77	NaN	NaN	NaN
610	9269.68	NaN	NaN	NaN
611	27225.34	NaN	NaN	NaN
612	2953.17	NaN	NaN	NaN
613	13853.36	NaN	NaN	NaN
614	19298.61	NaN	NaN	NaN
615	7869.47	NaN	NaN	NaN
616	13500.69	NaN	NaN	NaN
617	2573.48	NaN	NaN	NaN
618	31251.89	NaN	NaN	NaN
619	16095.32	NaN	NaN	NaN
621	38473.71	NaN	NaN	NaN
622	4472.47	NaN	NaN	NaN
623	23227.15	NaN	NaN	NaN
624	16924.74	NaN	NaN	NaN
625	23485.45	NaN	NaN	NaN
626	18621.12	NaN	NaN	NaN
627	13074.74	NaN	NaN	NaN
628	2539.57	NaN	NaN	NaN
629	7319.77	NaN	NaN	NaN
630	8734.58	NaN	NaN	NaN
631	7256.86	NaN	NaN	NaN
632	15383.35	NaN	NaN	NaN
633	11056.52	NaN	NaN	NaN
634	18165.16	NaN	NaN	NaN
635	3818.71	NaN	NaN	NaN
636	28498.91	NaN	NaN	NaN
637	35662.26	NaN	NaN	NaN
638	8242.62	NaN	NaN	NaN
640	14402.34	NaN	NaN	NaN
641	17334.11	NaN	NaN	NaN
642	8721.97	NaN	NaN	NaN
643	8156.11	NaN	NaN	NaN
644	53312.81	NaN	NaN	NaN
645	10740.09	NaN	NaN	NaN
646	3450.69	NaN	NaN	NaN
647	2625.10	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
648	21174.10	NaN	NaN	NaN
649	11147.91	NaN	NaN	NaN
650	8236.36	NaN	NaN	NaN
651	3026.25	NaN	NaN	NaN
654	4032.10	NaN	NaN	NaN
655	7814.16	NaN	NaN	NaN
656	35039.40	NaN	NaN	NaN
657	32481.05	NaN	NaN	NaN
659	3621.08	NaN	NaN	NaN
660	7809.13	NaN	NaN	NaN
661	10303.21	NaN	NaN	NaN
662	9232.79	NaN	NaN	NaN
663	2613.97	NaN	NaN	NaN
664	4590.49	NaN	NaN	NaN
666	65723.62	NaN	NaN	NaN
667	49221.63	NaN	NaN	NaN
668	8270.38	NaN	NaN	NaN
669	11477.47	NaN	NaN	NaN
670	33614.62	NaN	NaN	NaN
671	2886.35	NaN	NaN	NaN
672	5679.39	NaN	NaN	NaN
673	22213.53	NaN	NaN	NaN
675	13476.90	NaN	NaN	NaN
676	6861.84	NaN	NaN	NaN
677	9631.08	NaN	NaN	NaN
678	4569.15	NaN	NaN	NaN
679	25864.65	NaN	NaN	NaN
680	21493.44	NaN	NaN	NaN
683	32771.45	NaN	NaN	NaN
684	18942.34	NaN	NaN	NaN
685	12593.00	NaN	NaN	NaN
686	23576.99	NaN	NaN	NaN
688	24572.28	NaN	NaN	NaN
689	4524.77	NaN	NaN	NaN
690	11706.93	NaN	NaN	NaN
691	8467.17	NaN	NaN	NaN
692	8471.55	NaN	NaN	NaN
693	12354.16	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
694	8131.92	NaN	NaN	NaN
695	11178.46	NaN	NaN	NaN
697	10031.96	NaN	NaN	NaN
698	30480.85	NaN	NaN	NaN
699	4255.62	NaN	NaN	NaN
700	25640.67	NaN	NaN	NaN
701	7996.00	NaN	NaN	NaN
702	7011.74	NaN	NaN	NaN
704	6891.99	NaN	NaN	NaN
706	61639.73	NaN	NaN	NaN
707	2725.40	NaN	NaN	NaN
708	11967.39	NaN	NaN	NaN
709	34446.96	NaN	NaN	NaN
711	19676.51	NaN	NaN	NaN
712	8530.54	NaN	NaN	NaN
713	4004.65	NaN	NaN	NaN
715	10220.74	NaN	NaN	NaN
716	11425.01	NaN	NaN	NaN
717	4172.84	NaN	NaN	NaN
718	8041.62	NaN	NaN	NaN
719	11391.95	NaN	NaN	NaN
721	11431.35	NaN	NaN	NaN
723	6009.63	NaN	NaN	NaN
724	44312.98	NaN	NaN	NaN
725	15742.41	NaN	NaN	NaN
726	45664.45	NaN	NaN	NaN
727	10810.62	NaN	NaN	NaN
729	9817.06	NaN	NaN	NaN
730	10696.56	NaN	NaN	NaN
731	16027.02	NaN	NaN	NaN
732	19309.90	NaN	NaN	NaN
733	11601.33	NaN	NaN	NaN
735	10337.11	NaN	NaN	NaN
737	37704.71	NaN	NaN	NaN
738	20216.67	NaN	NaN	NaN
739	22213.53	NaN	NaN	NaN
740	6658.73	NaN	NaN	NaN
741	19855.31	NaN	NaN	NaN

	Prediction	Target	Residual	Difference%
742	16960.31	NaN	NaN	NaN
743	5268.17	NaN	NaN	NaN
744	2379.58	NaN	NaN	NaN
746	13355.11	NaN	NaN	NaN
748	48699.98	NaN	NaN	NaN
750	10381.62	NaN	NaN	NaN
751	8493.04	NaN	NaN	NaN
753	6358.55	NaN	NaN	NaN
754	17028.45	NaN	NaN	NaN
755	15885.66	NaN	NaN	NaN
756	3752.54	NaN	NaN	NaN
757	12028.91	NaN	NaN	NaN
759	10125.27	NaN	NaN	NaN
760	13443.32	NaN	NaN	NaN
761	9097.13	NaN	NaN	NaN
763	12383.35	NaN	NaN	NaN
764	14049.76	NaN	NaN	NaN
766	18982.15	NaN	NaN	NaN
767	24323.48	NaN	NaN	NaN
768	38260.36	NaN	NaN	NaN
770	10732.07	NaN	NaN	NaN
771	13922.45	NaN	NaN	NaN
772	27487.75	NaN	NaN	NaN
773	13491.16	NaN	NaN	NaN

In [ ]: