```
pip install ucimlrepo
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.6-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.6
```

```
!pip install hvplot
```

```
Collecting hvplot
  Downloading hvplot-0.9.2-py2.py3-none-any.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 17.4 MB/s eta 0:00:00
Requirement already satisfied: bokeh>=1.0.0 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: colorcet>=2 in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: holoviews>=1.11.0 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from h
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: panel>=0.11.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: param<3.0,>=1.12.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: contourpy>=1 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: xyzservices>=2021.09.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: pyviz-comms>=0.7.4 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: markdown in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: markdown-it-py in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: linkify-it-py in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: mdit-py-plugins in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: tqdm>=4.48.0 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from p
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: uc-micro-py in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dis
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-pack
Installing collected packages: hvplot
Successfully installed hvplot-0.9.2
```

```
from ucimlrepo import fetch_ucirepo
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import hvplot.pandas

from sklearn.model_selection import train_test_split

from sklearn import metrics

from sklearn.linear_model import LinearRegression

%matplotlib inline

# fetch dataset
automobile = fetch_ucirepo(id=10)

# data (as pandas dataframes)
X = automobile.data.features
y = automobile.data.targets

# metadata
#print(automobile.metadata)

# variable information
#print(automobile.variables)

df = pd.concat([X,y], axis=1)
```

```
df
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	fuel-system	en
0	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	
1	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	
2	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	mpfi	
3	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	mpfi	
4	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	mpfi	
...	...	...	...	...	...	...	...	...	...	
200	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	mpfi	
201	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	mpfi	
202	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	mpfi	
203	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	idi	
204	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	mpfi	

205 rows × 26 columns

Data Cleaning

```
# Imputing the missing values
df.fillna(df.select_dtypes(np.number).mean(), inplace=True)
```

```
# There are no more missing values in the dataframe
df.isnull().sum()
```

```
price                0
highway-mpg          0
city-mpg             0
peak-rpm             0
horsepower           0
compression-ratio    0
stroke               0
bore                 0
fuel-system          0
engine-size          0
num-of-cylinders     0
engine-type          0
curb-weight          0
```

```
height               0
width                0
length               0
wheel-base           0
engine-location      0
drive-wheels         0
body-style           0
num-of-doors         0
aspiration           0
fuel-type            0
make                 0
normalized-losses    0
symboling            0
dtype: int64
```

```
# Coverting categorical types into a numerical type
from sklearn.preprocessing import LabelEncoder

categorical_columns = ['fuel-system','engine-type','engine-location','drive-wheels','body-st

label_encoder = LabelEncoder()
for column in categorical_columns:
    encoded_column_name = column + '_encoded'
    df[encoded_column_name] = label_encoder.fit_transform(df[column])
```

```
df
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	fuel-system	en
0	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	
1	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	
2	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	mpfi	
3	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	mpfi	
4	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	mpfi	
...	...	...	...	...	...	...	...	...	...	
200	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	mpfi	
201	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	mpfi	
202	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	mpfi	
203	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	idi	
204	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	mpfi	

205 rows × 34 columns

```
# Dropping the categorical type in the dataframe for accurat linear regression analysis
categorical_columns = ['fuel-system','engine-type','engine-location','drive-wheels','body-st
df = df.drop(columns=categorical_columns,axis=1)
df
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	engine-size	c
0	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	130	
1	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	130	
2	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	152	
3	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	109	
4	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	136	
...	...	...	...	...	...	...	...	...	...	
200	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	141	
201	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	141	
202	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	173	
203	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	145	
204	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	141	

205 rows × 26 columns

```
df.dtypes
```

```
price                     float64
highway-mpg                 int64
city-mpg                    int64
peak-rpm                  float64
horsepower                float64
compression-ratio         float64
stroke                    float64
bore                      float64
engine-size                 int64
num-of-cylinders            int64
curb-weight                 int64
height                    float64
width                     float64
length                    float64
wheel-base                float64
num-of-doors              float64
normalized-losses         float64
symboling                   int64
fuel-system_encoded         int64
engine-type_encoded         int64
```

```
engine-location_encoded      int64
drive-wheels_encoded         int64
body-style_encoded           int64
aspiration_encoded           int64
fuel-type_encoded            int64
make_encoded                 int64
dtype: object
```

```
df.shape
```

```
(205, 26)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   price                    205 non-null    float64
 1   highway-mpg              205 non-null    int64
 2   city-mpg                 205 non-null    int64
 3   peak-rpm                 205 non-null    float64
 4   horsepower               205 non-null    float64
 5   compression-ratio        205 non-null    float64
 6   stroke                   205 non-null    float64
 7   bore                     205 non-null    float64
 8   engine-size              205 non-null    int64
 9   num-of-cylinders         205 non-null    int64
 10  curb-weight              205 non-null    int64
 11  height                   205 non-null    float64
 12  width                    205 non-null    float64
 13  length                   205 non-null    float64
 14  wheel-base               205 non-null    float64
 15  num-of-doors             205 non-null    float64
 16  normalized-losses        205 non-null    float64
 17  symboling                205 non-null    int64
 18  fuel-system_encoded      205 non-null    int64
 19  engine-type_encoded      205 non-null    int64
 20  engine-location_encoded  205 non-null    int64
 21  drive-wheels_encoded     205 non-null    int64
 22  body-style_encoded       205 non-null    int64
 23  aspiration_encoded       205 non-null    int64
 24  fuel-type_encoded        205 non-null    int64
 25  make_encoded             205 non-null    int64
dtypes: float64(12), int64(14)
memory usage: 41.8 KB
```

```
df.corr()
```

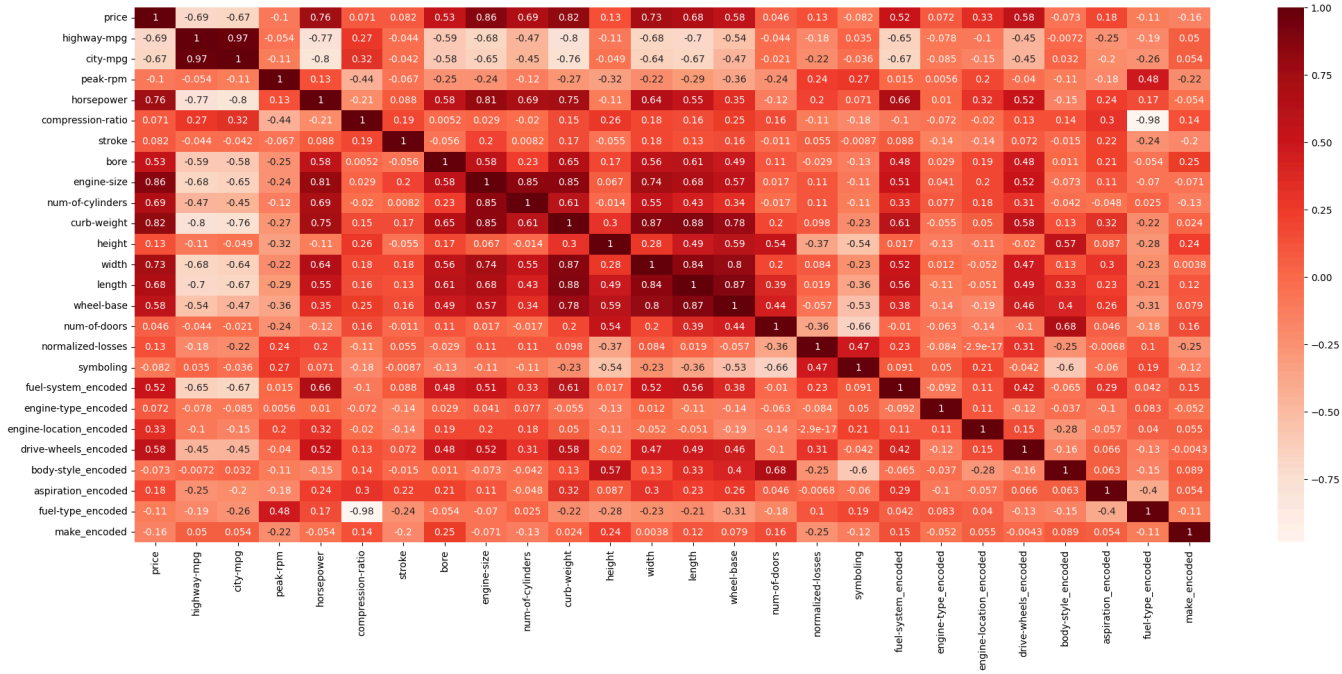
	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	
price	1.000000	-0.690526	-0.667449	-0.100854	0.757917	0.070990	0.
highway-mpg	-0.690526	1.000000	0.971337	-0.054257	-0.770903	0.265201	-0.
city-mpg	-0.667449	0.971337	1.000000	-0.113723	-0.803162	0.324701	-0.
peak-rpm	-0.100854	-0.054257	-0.113723	1.000000	0.130971	-0.435936	-0.
horsepower	0.757917	-0.770903	-0.803162	0.130971	1.000000	-0.205740	0.
compression-ratio	0.070990	0.265201	0.324701	-0.435936	-0.205740	1.000000	0.
stroke	0.082095	-0.043961	-0.042179	-0.066844	0.088264	0.186105	1.
bore	0.532300	-0.586992	-0.584508	-0.254761	0.575737	0.005201	-0.
engine-size	0.861752	-0.677470	-0.653658	-0.244599	0.810713	0.028971	0.
num-of-cylinders	0.687770	-0.466666	-0.445837	-0.124358	0.691208	-0.020002	0.
curb-weight	0.820825	-0.797465	-0.757414	-0.266283	0.750968	0.151362	0.
height	0.134388	-0.107358	-0.048640	-0.320602	-0.110137	0.261214	-0.
width	0.728699	-0.677218	-0.642704	-0.219859	0.642195	0.181129	0.
length	0.682986	-0.704662	-0.670909	-0.287031	0.554434	0.158414	0.
wheel-base	0.583168	-0.544082	-0.470414	-0.360704	0.351957	0.249786	0.
num-of-doors	0.046001	-0.044213	-0.020671	-0.240295	-0.124001	0.161502	-0.
normalized-losses	0.133999	-0.178221	-0.218749	0.237748	0.203434	-0.114525	0.
symboling	-0.082201	0.034606	-0.035823	0.273679	0.071389	-0.178515	-0.
fuel-system_encoded	0.516533	-0.645659	-0.671581	0.014714	0.659120	-0.100786	0.
engine-type_encoded	0.071535	-0.078456	-0.085004	0.005592	0.010258	-0.071873	-0.
engine-location_encoded	0.331013	-0.102026	-0.153487	0.198400	0.317610	-0.019762	-0.
drive-wheels_encoded	0.576866	-0.452220	-0.449581	-0.039724	0.516936	0.127479	0.
body-style_encoded	-0.072677	-0.007170	0.031697	-0.109410	-0.152438	0.136243	-0.

aspiration_encoded	0.177285	-0.254416	-0.202362	-0.183629	0.240182	0.295541	0.
fuel-type_encoded	-0.110207	-0.191392	-0.255963	0.477060	0.165190	-0.984356	-0.
make_encoded	-0.161471	0.050022	0.053642	-0.218342	-0.053654	0.138828	-0.

26 rows × 26 columns

```
fig, ax = plt.subplots(figsize = (25, 10))
sns.heatmap(df.corr(), annot=True, cmap='Reds')
```

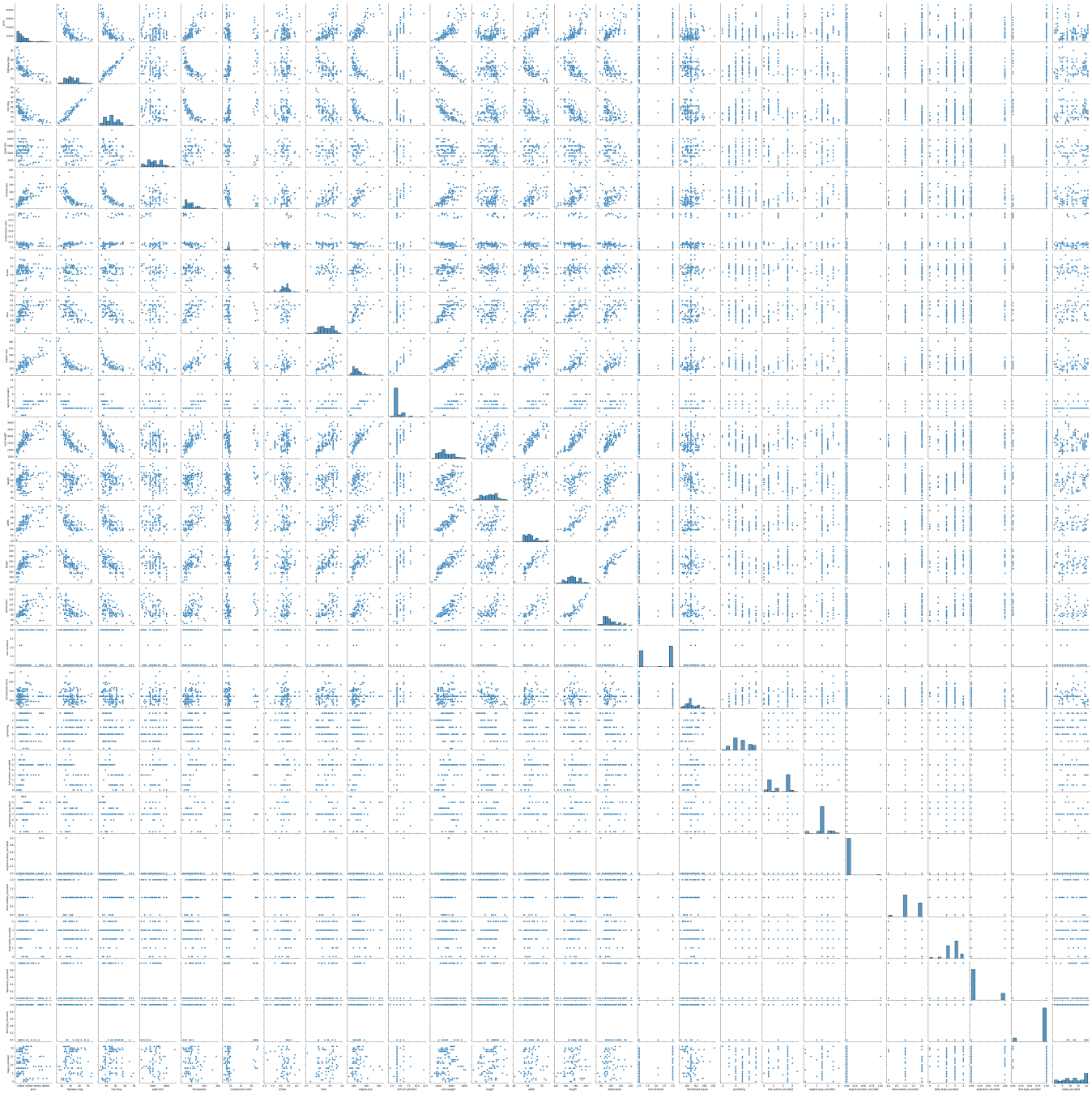
```
<Axes: >
```



Exploratory Data Analysis

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7c87ec280a00>
```



Training a Linear Regression Model

X and y arrays

```
X = df.drop('price',axis=1)

y = df['normalized-losses']
```

```
print("X-", X.shape, "\ny-", y.shape)
```

```
X- (205, 25)
y- (205,)
```

Train Test Split

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 10
```

```
X_train.shape
```

```
(143, 25)
```

```
X_test.shape
```

```
(62, 25)
```

Linear Regression

```
model = LinearRegression()
```

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

```
▾ LinearRegression
LinearRegression()
```

Model Evaluation

```
model.coef_
```

```
array([-2.78263758e-15,  1.47624968e-15, -1.09236736e-16,  2.71685639e-16,
       -1.31928379e-15, -8.49612120e-15,  3.35728726e-14,  4.75795023e-16,
        1.95259003e-15, -1.54871236e-16,  1.38015749e-15, -5.92406309e-16,
       -4.87730338e-16, -6.98331142e-16,  1.11063094e-14,  1.00000000e+00,
        1.32510326e-15, -5.11506256e-16, -3.98484956e-16, -3.35745331e-16,
        2.43758464e-15,  1.28467343e-15,  1.33107585e-15, -2.04532711e-14,
        1.31877948e-16])
```

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

```
▾ LinearRegression
LinearRegression()
```

```
model.coef_
```

```
array([-2.78263758e-15,  1.47624968e-15, -1.09236736e-16,  2.71685639e-16,
       -1.31928379e-15, -8.49612120e-15,  3.35728726e-14,  4.75795023e-16,
        1.95259003e-15, -1.54871236e-16,  1.38015749e-15, -5.92406309e-16,
       -4.87730338e-16, -6.98331142e-16,  1.11063094e-14,  1.00000000e+00,
        1.32510326e-15, -5.11506256e-16, -3.98484956e-16, -3.35745331e-16,
        2.43758464e-15,  1.28467343e-15,  1.33107585e-15, -2.04532711e-14,
        1.31877948e-16])
```

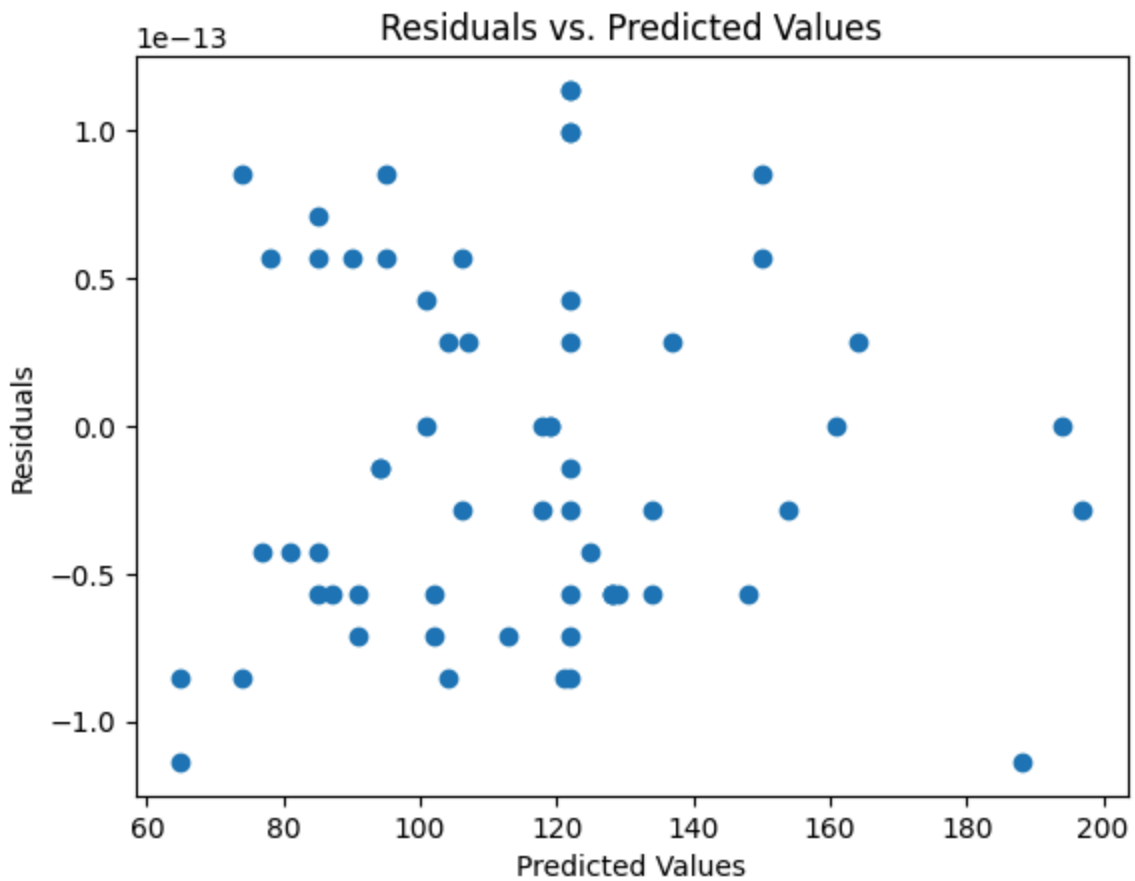
```
pd.DataFrame(model.coef_, X.columns, columns=['Coeffiecients'])
```

	Coeffiecients
highway-mpg	-2.782638e-15
city-mpg	1.476250e-15
peak-rpm	-1.092367e-16
horsepower	2.716856e-16
compression-ratio	-1.319284e-15
stroke	-8.496121e-15
bore	3.357287e-14
engine-size	4.757950e-16
num-of-cylinders	1.952590e-15
curb-weight	-1.548712e-16
height	1.380157e-15
width	-5.924063e-16
length	-4.877303e-16
wheel-base	-6.983311e-16
num-of-doors	1.110631e-14
normalized-losses	1.000000e+00
symboling	1.325103e-15
fuel-system_encoded	-5.115063e-16
engine-type_encoded	-3.984850e-16
engine-location_encoded	-3.357453e-16
drive-wheels_encoded	2.437585e-15
body-style_encoded	1.284673e-15
aspiration_encoded	1.331076e-15
fuel-type_encoded	-2.045327e-14
make_encoded	1.318779e-16

Prediction from our Model

```
y_pred = model.predict(X_test)
residuals = y_test - y_pred

plt.scatter(y_pred, residuals)
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residuals vs. Predicted Values')
plt.show()
```



Regression Evaluation Metrics

```
MAE = metrics.mean_absolute_error(y_test, y_pred)
MSE = metrics.mean_squared_error(y_test,y_pred)
RMSE = np.sqrt(MSE)
```

```
MAE
```

```
5.271768684671711e-14
```

```
MSE
```

```
3.713244622253227e-27
```

```
RMSE
```

```
6.093639817262937e-14
```

```
df['normalized-losses'].mean()
```

```
122.0
```

Residual Analysis

```
test_residual = y_test - y_pred
```

```
pd.DataFrame({'Error Values': (test_residual)}).hvplot.kde()
```

```
/usr/local/lib/python3.10/dist-packages/holoviews/core/util.py:1585: PanelDeprecationWar
  value = param_value_if_widget(value)
```

```
sns.displot(test_residual, bins=25, kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7c87c2a74310>
```