Linear Regression on Automobiles

In [ ]:
```
!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
```

In [ ]:
```
#importing libaries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
from scipy import stats
```

In [ ]:
```
from ucimlrepo import fetch_ucirepo

# 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)
```

```
{'uci_id': 10, 'name': 'Automobile', 'repository_url': 'https://archive.ics.uci.edu/
dataset/10/automobile', 'data_url': 'https://archive.ics.uci.edu/static/public/10/da
ta.csv', 'abstract': "From 1985 Ward's Automotive Yearbook", 'area': 'Other', 'task
s': ['Regression'], 'characteristics': ['Multivariate'], 'num_instances': 205, 'num_
features': 25, 'feature_types': ['Categorical', 'Integer', 'Real'], 'demographics':
[], 'target_col': ['symboling'], 'index_col': None, 'has_missing_values': 'yes', 'mi
ssing_values_symbol': 'NaN', 'year_of_dataset_creation': 1985, 'last_updated': 'Thu
Aug 10 2023', 'dataset_doi': '10.24432/C5B01C', 'creators': ['Jeffrey Schlimmer'],
'intro_paper': None, 'additional_info': {'summary': 'This data set consists of three
types of entities: (a) the specification of an auto in terms of various characterist
ics, (b) its assigned insurance risk rating, (c) its normalized losses in use as com
pared to other cars.  The second rating corresponds to the degree to which the auto
is more risky than its price indicates. Cars are initially assigned a risk factor sy
mbol associated with its price.   Then, if it is more risky (or less), this symbol i
s adjusted by moving it up (or down) the scale.  Actuarians call this process "symbo
ling".  A value of +3 indicates that the auto is risky, -3 that it is probably prett
y safe.\r\n\r\nThe third factor is the relative average loss payment per insured veh
icle year.  This value is normalized for all autos within a particular size classifi
cation (two-door small, station wagons, sports/speciality, etc...), and represents t
he average loss per car per year.\r\n\r\nNote: Several of the attributes in the data
base could be used as a "class" attribute.', 'purpose': None, 'funded_by': None, 'in
stances_represent': None, 'recommended_data_splits': None, 'sensitive_data': None,
'preprocessing_description': None, 'variable_info': 'Attribute: Attribute Range\r\n
\r\n  1. symboling:                -3, -2, -1, 0, 1, 2, 3.\r\n  2. normalized-losse
s:        continuous from 65 to 256.\r\n  3. make:                     \r\n
alfa-romero, audi, bmw, chevrolet, dodge, honda,\r\n                               i
suzu, jaguar, mazda, mercedes-benz, mercury,\r\n                               mitsu
bishi, nissan, peugot, plymouth, porsche,\r\n                               renault,
saab, subaru, toyota, volkswagen, volvo\r\n\r\n  4. fuel-type:                diese
l, gas.\r\n  5. aspiration:              std, turbo.\r\n  6. num-of-doors:
four, two.\r\n  7. body-style:               hardtop, wagon, sedan, hatchback, conve
rtible.\r\n  8. drive-wheels:            4wd, fwd, rwd.\r\n  9. engine-location:
front, rear.\r\n 10. wheel-base:               continuous from 86.6 120.9.\r\n 11. l
ength:                   continuous from 141.1 to 208.1.\r\n 12. width:
continuous from 60.3 to 72.3.\r\n 13. height:                  continuous from 47.8
to 59.8.\r\n 14. curb-weight:              continuous from 1488 to 4066.\r\n 15. eng
ine-type:              dohc, dohcv, l, ohc, ohcf, ohcv, rotor.\r\n 16. num-of-cylind
ers:         eight, five, four, six, three, twelve, two.\r\n 17. engine-size:
continuous from 61 to 326.\r\n 18. fuel-system:              1bbl, 2bbl, 4bbl, idi,
mfi, mpfi, spdi, spfi.\r\n 19. bore:                     continuous from 2.54 to 3.9
4.\r\n 20. stroke:                   continuous from 2.07 to 4.17.\r\n 21. compressi
on-ratio:        continuous from 7 to 23.\r\n 22. horsepower:              continuo
us from 48 to 288.\r\n 23. peak-rpm:                 continuous from 4150 to 6600.\r
\n 24. city-mpg:                   continuous from 13 to 49.\r\n 25. highway-mpg:
continuous from 16 to 54.\r\n 26. price:                    continuous from 5118 to
45400.', 'citation': None}}
```


```
                 name     role         type demographic  \
0               price  Feature   Continuous        None
1         highway-mpg  Feature   Continuous        None
2            city-mpg  Feature   Continuous        None
3            peak-rpm  Feature   Continuous        None
4          horsepower  Feature   Continuous        None
5   compression-ratio  Feature   Continuous        None
6              stroke  Feature   Continuous        None
7                bore  Feature   Continuous        None
8         fuel-system  Feature  Categorical        None
```

```
9          engine-size  Feature   Continuous        None
10    num-of-cylinders  Feature      Integer        None
11         engine-type  Feature  Categorical        None
12         curb-weight  Feature   Continuous        None
13              height  Feature   Continuous        None
14               width  Feature   Continuous        None
15              length  Feature   Continuous        None
16          wheel-base  Feature   Continuous        None
17     engine-location  Feature       Binary        None
18        drive-wheels  Feature  Categorical        None
19          body-style  Feature  Categorical        None
20         num-of-doors  Feature      Integer        None
21          aspiration  Feature       Binary        None
22           fuel-type  Feature       Binary        None
23                make  Feature  Categorical        None
24   normalized-losses  Feature   Continuous        None
25           symboling   Target      Integer        None

                                         description units missing_values
0                   continuous from 5118 to 45400  None            yes
1                       continuous from 16 to 54  None             no
2                       continuous from 13 to 49  None             no
3                    continuous from 4150 to 6600  None            yes
4                      continuous from 48 to 288  None            yes
5                        continuous from 7 to 23  None             no
6                    continuous from 2.07 to 4.17  None            yes
7                    continuous from 2.54 to 3.94  None            yes
8         1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi  None       no
9                      continuous from 61 to 326  None             no
10          eight, five, four, six, three, twelve, two  None       no
11              dohc, dohcv, l, ohc, ohcf, ohcv, rotor  None       no
12                 continuous from 1488 to 4066  None              no
13                 continuous from 47.8 to 59.8  None              no
14                 continuous from 60.3 to 72.3  None              no
15               continuous from 141.1 to 208.1  None              no
16                  continuous from 86.6 120.9  None               no
17                                  front, rear  None              no
18                                4wd, fwd, rwd  None              no
19      hardtop, wagon, sedan, hatchback, convertible  None        no
20                                    four, two  None             yes
21                                   std, turbo  None              no
22                                  diesel, gas  None              no
23  alfa-romero, audi, bmw, chevrolet, dodge, hond...  None        no
24                     continuous from 65 to 256  None            yes
25                          -3, -2, -1, 0, 1, 2, 3  None           no
```

In [ ]:
```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 25 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   price              201 non-null    float64
 1   highway-mpg        205 non-null    int64  
 2   city-mpg           205 non-null    int64  
 3   peak-rpm           203 non-null    float64
 4   horsepower         203 non-null    float64
 5   compression-ratio  205 non-null    float64
 6   stroke             201 non-null    float64
 7   bore               201 non-null    float64
 8   fuel-system        205 non-null    object 
 9   engine-size        205 non-null    int64  
 10  num-of-cylinders   205 non-null    int64  
 11  engine-type        205 non-null    object 
 12  curb-weight        205 non-null    int64  
 13  height             205 non-null    float64
 14  width              205 non-null    float64
 15  length             205 non-null    float64
 16  wheel-base         205 non-null    float64
 17  engine-location    205 non-null    object 
 18  drive-wheels       205 non-null    object 
 19  body-style         205 non-null    object 
 20  num-of-doors       203 non-null    float64
 21  aspiration         205 non-null    object 
 22  fuel-type          205 non-null    object 
 23  make               205 non-null    object 
 24  normalized-losses  164 non-null    float64
dtypes: float64(12), int64(5), object(8)
memory usage: 40.2+ KB
```

In [ ]:

```
print(automobile.variables)
```

```
                 name     role         type demographic  \
0               price  Feature   Continuous        None   
1         highway-mpg  Feature   Continuous        None   
2            city-mpg  Feature   Continuous        None   
3            peak-rpm  Feature   Continuous        None   
4          horsepower  Feature   Continuous        None   
5   compression-ratio  Feature   Continuous        None   
6              stroke  Feature   Continuous        None   
7                bore  Feature   Continuous        None   
8         fuel-system  Feature  Categorical        None   
9         engine-size  Feature   Continuous        None   
10   num-of-cylinders  Feature      Integer        None   
11        engine-type  Feature  Categorical        None   
12        curb-weight  Feature   Continuous        None   
13             height  Feature   Continuous        None   
14              width  Feature   Continuous        None   
15             length  Feature   Continuous        None   
16         wheel-base  Feature   Continuous        None   
17    engine-location  Feature       Binary        None   
18       drive-wheels  Feature  Categorical        None   
19         body-style  Feature  Categorical        None   
20        num-of-doors  Feature      Integer        None   
21         aspiration  Feature       Binary        None   
22          fuel-type  Feature       Binary        None   
23               make  Feature  Categorical        None   
24  normalized-losses  Feature   Continuous        None   
25          symboling   Target      Integer        None   

                                     description units missing_values  
0                continuous from 5118 to 45400  None            yes  
1                     continuous from 16 to 54  None             no  
2                     continuous from 13 to 49  None             no  
3                 continuous from 4150 to 6600  None            yes  
4                    continuous from 48 to 288  None            yes  
5                      continuous from 7 to 23  None             no  
6                 continuous from 2.07 to 4.17  None            yes  
7                 continuous from 2.54 to 3.94  None            yes  
8       1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi  None             no  
9                    continuous from 61 to 326  None             no  
10         eight, five, four, six, three, twelve, two  None             no  
11              dohc, dohcv, l, ohc, ohcf, ohcv, rotor  None             no  
12                continuous from 1488 to 4066  None             no  
13                continuous from 47.8 to 59.8  None             no  
14                continuous from 60.3 to 72.3  None             no  
15              continuous from 141.1 to 208.1  None             no  
16                 continuous from 86.6 120.9  None             no  
17                                 front, rear  None             no  
18                               4wd, fwd, rwd  None             no  
19     hardtop, wagon, sedan, hatchback, convertible  None             no  
20                                   four, two  None            yes  
21                                  std, turbo  None             no  
22                                 diesel, gas  None             no  
23  alfa-romero, audi, bmw, chevrolet, dodge, hond...  None             no  
24                   continuous from 65 to 256  None            yes  
25                       -3, -2, -1, 0, 1, 2, 3  None             no  
```

In [ ]:
```
print(automobile.data.columns)
```

```
None
```

In [ ]:
```
from ucimlrepo import fetch_ucirepo

# Fetch dataset
automobile = fetch_ucirepo(id=10)

if automobile is None:
    print("Error: Dataset loading failed.")
else:
    print("Dataset loaded successfully.")
```

```
Dataset loaded successfully.
```

Loading the dataset

In [ ]:
```
automobile = fetch_ucirepo(id=10)
X = automobile.data.features
y = automobile.data.target
```

Data Wrangling

In [ ]:
```
#checks for missing values
print(new_X.isnull().sum())
```

```
price                     4
highway-mpg               0
city-mpg                  0
peak-rpm                  2
horsepower                2
                         ..
make_toyota               0
make_volkswagen           0
make_volvo                0
engine-location_front     0
engine-location_rear      0
Length: 68, dtype: int64
```

In [ ]:
```
#encodes categorical variables
new_X = pd.get_dummies(X, columns=['fuel-system', 'engine-type', 'drive-wheels', 'b

print(new_X.shape)
print(new_X.dtypes)
```

```
(205, 68)
price                     float64
highway-mpg                 int64
city-mpg                    int64
peak-rpm                  float64
horsepower                float64
                           ...
make_toyota                  bool
make_volkswagen              bool
make_volvo                   bool
engine-location_front        bool
engine-location_rear         bool
Length: 68, dtype: object
```

Feature Scaling:

In [ ]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

numeric_columns = new_X.select_dtypes(include=['float64', 'int64']).columns
X_scaled = new_X.copy()
X_scaled[numeric_columns] = scaler.fit_transform(new_X[numeric_columns])
```

Data Exploration

In [ ]:
```
X.describe()
```

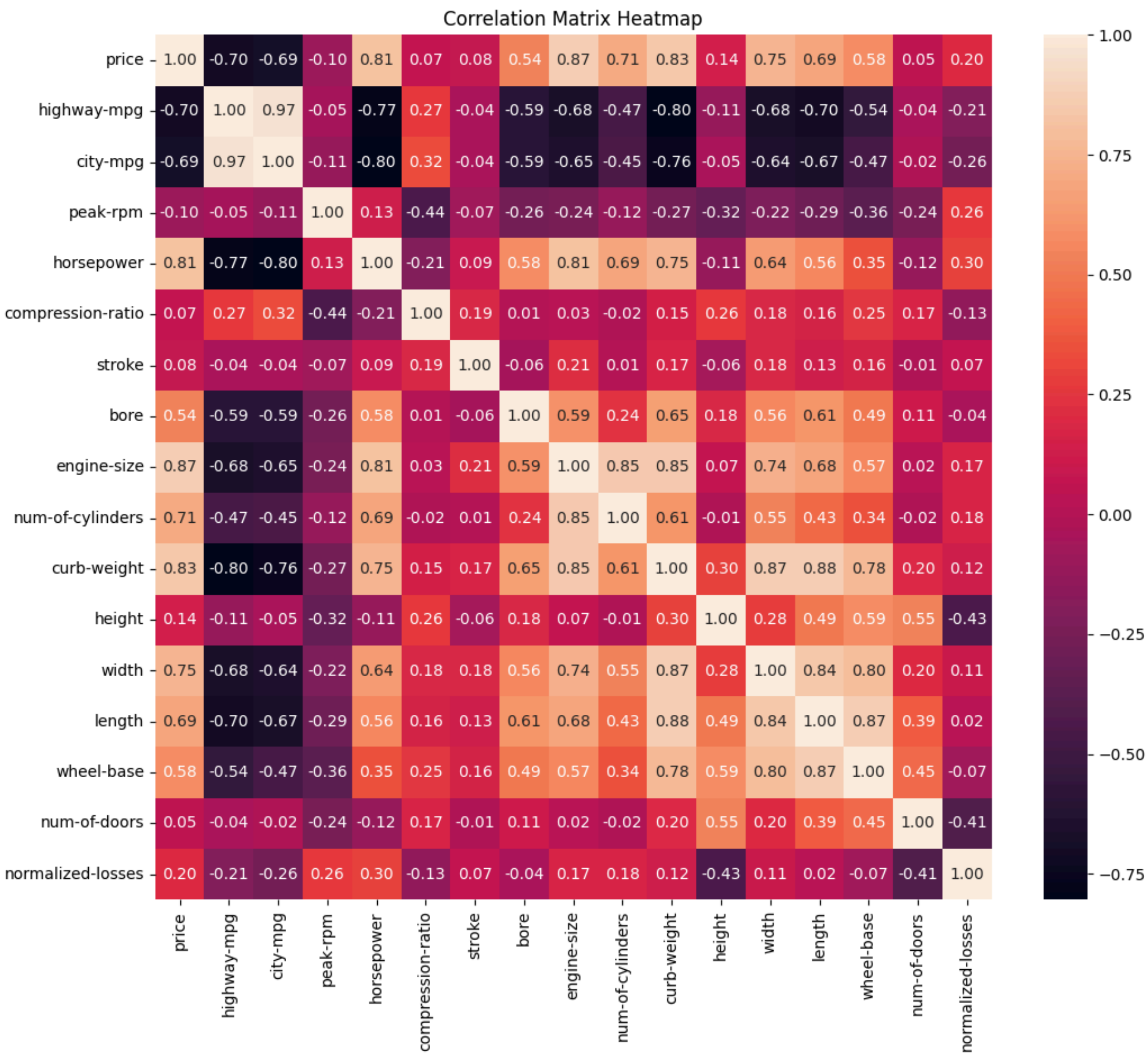
Out[ ]:

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	
count	201.000000	205.000000	205.000000	203.000000	203.000000	205.000000	201.(
mean	13207.129353	30.751220	25.219512	5125.369458	104.256158	10.142537	3.:
std	7947.066342	6.886443	6.542142	479.334560	39.714369	3.972040	0.:
min	5118.000000	16.000000	13.000000	4150.000000	48.000000	7.000000	2.(
25%	7775.000000	25.000000	19.000000	4800.000000	70.000000	8.600000	3.'
50%	10295.000000	30.000000	24.000000	5200.000000	95.000000	9.000000	3.:
75%	16500.000000	34.000000	30.000000	5500.000000	116.000000	9.400000	3.4
max	45400.000000	54.000000	49.000000	6600.000000	288.000000	23.000000	4.'

In [ ]:
```
#correlatiioin analysis
print(new_X.dtypes)

data_numeric = new_X.select_dtypes(include=['float64', 'int64'])

correlation_matrix = data_numeric.corr()
```

```
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='rocket', fmt=".2f")
plt.title("Correlation Matrix Heatmap")
plt.show()
```

```
price                      float64
highway-mpg                  int64
city-mpg                     int64
peak-rpm                   float64
horsepower                 float64
                            ...
make_toyota                   bool
make_volkswagen               bool
make_volvo                    bool
engine-location_front         bool
engine-location_rear          bool
Length: 68, dtype: object
```



	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	engine-size	num-of-cylinders	curb-weight	height	width	length	wheel-base	num-of-doors	normalized-losses
price	1.00	-0.70	-0.69	-0.10	0.81	0.07	0.08	0.54	0.87	0.71	0.83	0.14	0.75	0.69	0.58	0.05	0.20
highway-mpg	-0.70	1.00	0.97	-0.05	-0.77	0.27	-0.04	-0.59	-0.68	-0.47	-0.80	-0.11	-0.68	-0.70	-0.54	-0.04	-0.21
city-mpg	-0.69	0.97	1.00	-0.11	-0.80	0.32	-0.04	-0.59	-0.65	-0.45	-0.76	-0.05	-0.64	-0.67	-0.47	-0.02	-0.26
peak-rpm	-0.10	-0.05	-0.11	1.00	0.13	-0.44	-0.07	-0.26	-0.24	-0.12	-0.27	-0.32	-0.22	-0.29	-0.36	-0.24	0.26
horsepower	0.81	-0.77	-0.80	0.13	1.00	-0.21	0.09	0.58	0.81	0.69	0.75	-0.11	0.64	0.56	0.35	-0.12	0.30
compression-ratio	0.07	0.27	0.32	-0.44	-0.21	1.00	0.19	0.01	0.03	-0.02	0.15	0.26	0.18	0.16	0.25	0.17	-0.13
stroke	0.08	-0.04	-0.04	-0.07	0.09	0.19	1.00	-0.06	0.21	0.01	0.17	-0.06	0.18	0.13	0.16	-0.01	0.07
bore	0.54	-0.59	-0.59	-0.26	0.58	0.01	-0.06	1.00	0.59	0.24	0.65	0.18	0.56	0.61	0.49	0.11	-0.04
engine-size	0.87	-0.68	-0.65	-0.24	0.81	0.03	0.21	0.59	1.00	0.85	0.85	0.07	0.74	0.68	0.57	0.02	0.17
num-of-cylinders	0.71	-0.47	-0.45	-0.12	0.69	-0.02	0.01	0.24	0.85	1.00	0.61	-0.01	0.55	0.43	0.34	-0.02	0.18
curb-weight	0.83	-0.80	-0.76	-0.27	0.75	0.15	0.17	0.65	0.85	0.61	1.00	0.30	0.87	0.88	0.78	0.20	0.12
height	0.14	-0.11	-0.05	-0.32	-0.11	0.26	-0.06	0.18	0.07	-0.01	0.30	1.00	0.28	0.49	0.59	0.55	-0.43
width	0.75	-0.68	-0.64	-0.22	0.64	0.18	0.18	0.56	0.74	0.55	0.87	0.28	1.00	0.84	0.80	0.20	0.11
length	0.69	-0.70	-0.67	-0.29	0.56	0.16	0.13	0.61	0.68	0.43	0.88	0.49	0.84	1.00	0.87	0.39	0.02
wheel-base	0.58	-0.54	-0.47	-0.36	0.35	0.25	0.16	0.49	0.57	0.34	0.78	0.59	0.80	0.87	1.00	0.45	-0.07
num-of-doors	0.05	-0.04	-0.02	-0.24	-0.12	0.17	-0.01	0.11	0.02	-0.02	0.20	0.55	0.20	0.39	0.45	1.00	-0.41
normalized-losses	0.20	-0.21	-0.26	0.26	0.30	-0.13	0.07	-0.04	0.17	0.18	0.12	-0.43	0.11	0.02	-0.07	-0.41	1.00

Simple Linear Regression

Split Data into Training and Testing Sets:

In [ ]:
```
#splitting the data into training and testing sets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.2, ra
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-17-5ebc7f651068> in <cell line: 3>()
      1 from sklearn.model_selection import train_test_split
      2
----> 3 X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=
0.2, random_state=42)

NameError: name 'X_encoded' is not defined
```

In [ ]:
```
from sklearn.linear_model import LinearRegression

model = LinearRegression()

model.fit(X_train, y_train)
```

Out[ ]:
```
▾ LinearRegression
LinearRegression()
```

Fit Linear Regression Model:

In [ ]:
```
from sklearn.metrics import mean_squared_error, r2_score

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")
```

```
Mean Squared Error: 0.6859129086119733
R-squared: 0.5320537340191854
```

Logistic Regression on Wine

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
from ucimlrepo import fetch_ucirepo

# fetch dataset
wine = fetch_ucirepo(id=109)

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

# metadata
print(wine.metadata)

# variable information
print(wine.variables)
```

```
{'uci_id': 109, 'name': 'Wine', 'repository_url': 'https://archive.ics.uci.edu/datas
et/109/wine', 'data_url': 'https://archive.ics.uci.edu/static/public/109/data.csv',
'abstract': 'Using chemical analysis to determine the origin of wines', 'area': 'Phy
sics and Chemistry', 'tasks': ['Classification'], 'characteristics': ['Tabular'], 'n
um_instances': 178, 'num_features': 13, 'feature_types': ['Integer', 'Real'], 'demog
raphics': [], 'target_col': ['class'], 'index_col': None, 'has_missing_values': 'n
o', 'missing_values_symbol': None, 'year_of_dataset_creation': 1992, 'last_updated':
'Mon Aug 28 2023', 'dataset_doi': '10.24432/C5PC7J', 'creators': ['Stefan Aeberhar
d', 'M. Forina'], 'intro_paper': {'title': 'Comparative analysis of statistical patt
ern recognition methods in high dimensional settings', 'authors': 'S. Aeberhard, D.
Coomans, O. Vel', 'published_in': 'Pattern Recognition', 'year': 1994, 'url': 'http
s://www.semanticscholar.org/paper/83dc3e4030d7b9fbdbb4bde03ce12ab70ca10528', 'doi':
'10.1016/0031-3203(94)90145-7'}, 'additional_info': {'summary': 'These data are the
results of a chemical analysis of wines grown in the same region in Italy but derive
d from three different cultivars. The analysis determined the quantities of 13 const
ituents found in each of the three types of wines. \r\n\r\nI think that the initial
data set had around 30 variables, but for some reason I only have the 13 dimensional
version. I had a list of what the 30 or so variables were, but a.)  I lost it, and
b.), I would not know which 13 variables are included in the set.\r\n\r\nThe attribu
tes are (dontated by Riccardo Leardi, riclea@anchem.unige.it )\r\n1) Alcohol\r\n2) M
alic acid\r\n3) Ash\r\n4) Alcalinity of ash  \r\n5) Magnesium\r\n6) Total phenols\r
\n7) Flavanoids\r\n8) Nonflavanoid phenols\r\n9) Proanthocyanins\r\n10)Color intensi
ty\r\n11)Hue\r\n12)OD280/OD315 of diluted wines\r\n13)Proline \r\n\r\nIn a classific
ation context, this is a well posed problem with "well behaved" class structures. A
good data set for first testing of a new classifier, but not very challenging.
', 'purpose': 'test', 'funded_by': None, 'instances_represent': None, 'recommended_d
ata_splits': None, 'sensitive_data': None, 'preprocessing_description': None, 'varia
ble_info': 'All attributes are continuous\r\n\t\r\nNo statistics available, but sugg
est to standardise variables for certain uses (e.g. for us with classifiers which ar
e NOT scale invariant)\r\n\r\nNOTE: 1st attribute is class identifier (1-3)', 'citat
ion': None}}
```


```
                            name     role         type demographic  \
0                          class   Target  Categorical        None   
1                        Alcohol  Feature   Continuous        None   
2                      Malicacid  Feature   Continuous        None   
3                            Ash  Feature   Continuous        None   
4              Alcalinity_of_ash  Feature   Continuous        None   
5                      Magnesium  Feature      Integer        None   
6                  Total_phenols  Feature   Continuous        None   
7                     Flavanoids  Feature   Continuous        None   
8           Nonflavanoid_phenols  Feature   Continuous        None   
9                Proanthocyanins  Feature   Continuous        None   
10               Color_intensity  Feature   Continuous        None   
11                           Hue  Feature   Continuous        None   
12  0D280_0D315_of_diluted_wines  Feature   Continuous        None   
13                       Proline  Feature      Integer        None   

   description units missing_values  
0         None  None             no  
1         None  None             no  
2         None  None             no  
3         None  None             no  
4         None  None             no  
5         None  None             no  
6         None  None             no  
7         None  None             no  
```

```
8          None   None              no
9          None   None              no
10         None   None              no
11         None   None              no
12         None   None              no
13         None   None              no
```

Loading the dataset

In [ ]:
```
wine = pd.DataFrame(data=X, columns=wine.variables['name'][1:])
wine['class'] = y
```

Data Exploration

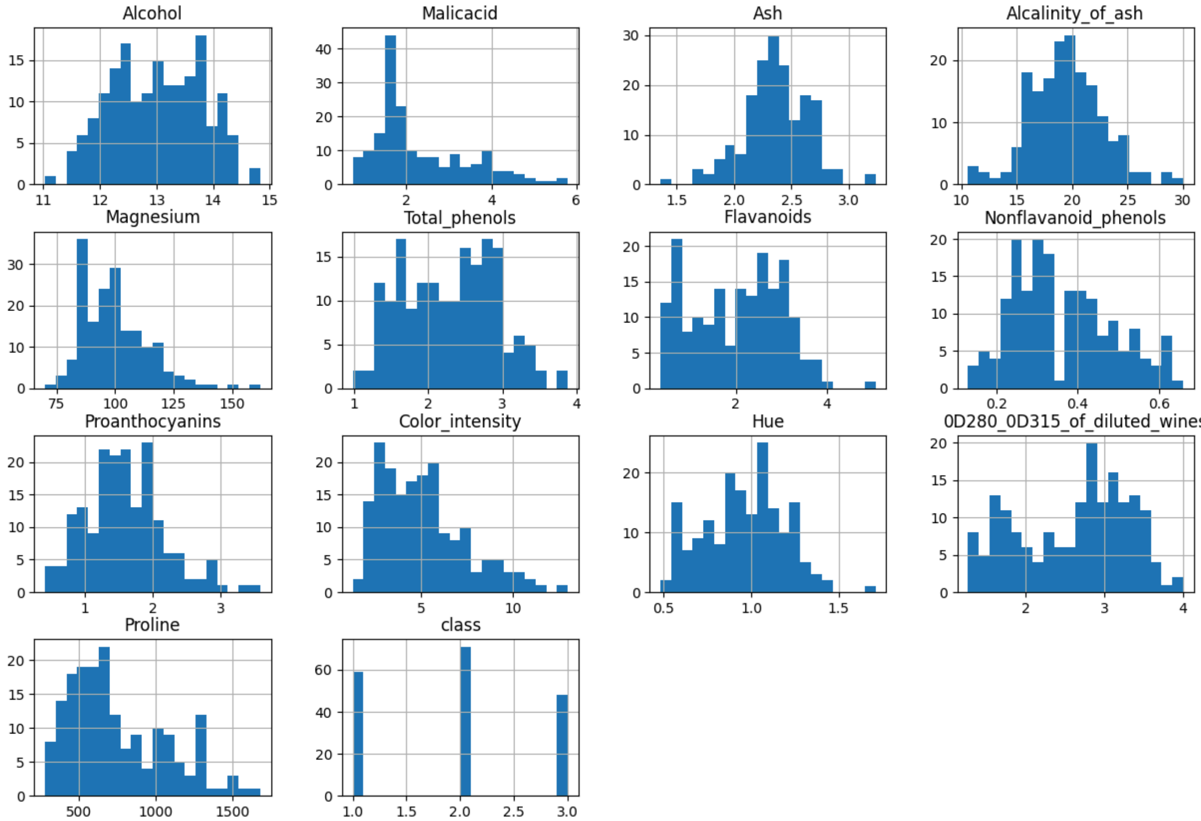
In [ ]:
```
print(wine.head())
print(wine.describe())
print(wine.info())
```

```
name  Alcohol  Malicacid   Ash  Alcalinity_of_ash  Magnesium  Total_phenols  \
0       14.23       1.71  2.43               15.6        127           2.80
1       13.20       1.78  2.14               11.2        100           2.65
2       13.16       2.36  2.67               18.6        101           2.80
3       14.37       1.95  2.50               16.8        113           3.85
4       13.24       2.59  2.87               21.0        118           2.80

name  Flavanoids  Nonflavanoid_phenols  Proanthocyanins  Color_intensity  \
0           3.06                  0.28             2.29             5.64
1           2.76                  0.26             1.28             4.38
2           3.24                  0.30             2.81             5.68
3           3.49                  0.24             2.18             7.80
4           2.69                  0.39             1.82             4.32

name   Hue  0D280_0D315_of_diluted_wines  Proline  class
0     1.04                          3.92     1065      1
1     1.05                          3.40     1050      1
2     1.03                          3.17     1185      1
3     0.86                          3.45     1480      1
4     1.04                          2.93      735      1
name      Alcohol   Malicacid         Ash  Alcalinity_of_ash   Magnesium  \
count  178.000000  178.000000  178.000000         178.000000  178.000000
mean    13.000618    2.336348    2.366517          19.494944   99.741573
std      0.811827    1.117146    0.274344           3.339564   14.282484
min     11.030000    0.740000    1.360000          10.600000   70.000000
25%     12.362500    1.602500    2.210000          17.200000   88.000000
50%     13.050000    1.865000    2.360000          19.500000   98.000000
75%     13.677500    3.082500    2.557500          21.500000  107.000000
max     14.830000    5.800000    3.230000          30.000000  162.000000

name   Total_phenols  Flavanoids  Nonflavanoid_phenols  Proanthocyanins  \
count     178.000000  178.000000            178.000000       178.000000
mean        2.295112    2.029270              0.361854         1.590899
std         0.625851    0.998859              0.124453         0.572359
min         0.980000    0.340000              0.130000         0.410000
25%         1.742500    1.205000              0.270000         1.250000
50%         2.355000    2.135000              0.340000         1.555000
75%         2.800000    2.875000              0.437500         1.950000
max         3.880000    5.080000              0.660000         3.580000

name   Color_intensity         Hue  0D280_0D315_of_diluted_wines      Proline  \
count       178.000000  178.000000                    178.000000   178.000000
mean          5.058090    0.957449                      2.611685   746.893258
std           2.318286    0.228572                      0.709990   314.907474
min           1.280000    0.480000                      1.270000   278.000000
25%           3.220000    0.782500                      1.937500   500.500000
50%           4.690000    0.965000                      2.780000   673.500000
75%           6.200000    1.120000                      3.170000   985.000000
max          13.000000    1.710000                      4.000000  1680.000000

name        class
count  178.000000
mean     1.938202
std      0.775035
min      1.000000
25%      1.000000
```

```
50%      2.000000
75%      3.000000
max      3.000000
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Alcohol                       178 non-null    float64
 1   Malicacid                     178 non-null    float64
 2   Ash                           178 non-null    float64
 3   Alcalinity_of_ash             178 non-null    float64
 4   Magnesium                     178 non-null    int64  
 5   Total_phenols                 178 non-null    float64
 6   Flavanoids                    178 non-null    float64
 7   Nonflavanoid_phenols          178 non-null    float64
 8   Proanthocyanins               178 non-null    float64
 9   Color_intensity               178 non-null    float64
 10  Hue                           178 non-null    float64
 11  0D280_0D315_of_diluted_wines  178 non-null    float64
 12  Proline                       178 non-null    int64  
 13  class                         178 non-null    int64  
dtypes: float64(11), int64(3)
memory usage: 19.6 KB
None
```

Data Visualization

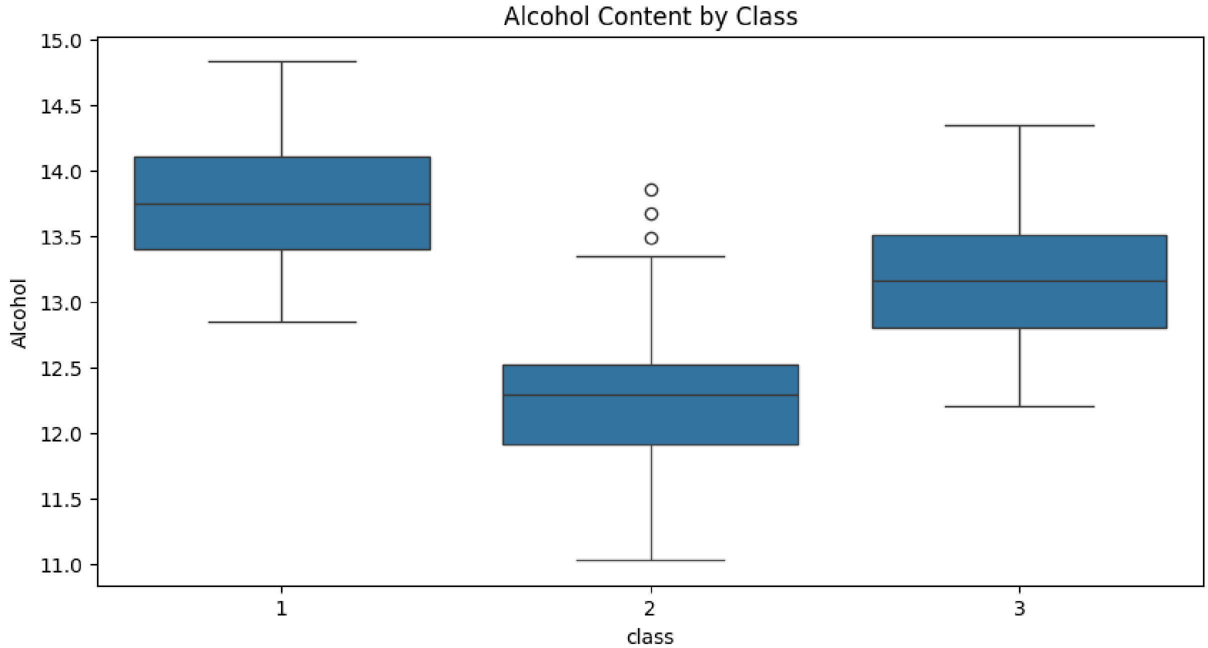
In [ ]:
```
wine.hist(bins=20, figsize=(15, 10))
plt.show()
```



visualize relationships between features

In [ ]:
```
sns.pairplot(wine, hue='class')
plt.show()
```

In [ ]:
```
plt.figure(figsize=(10, 5))
sns.boxplot(x='class', y='Alcohol', data=wine)
plt.title('Alcohol Content by Class')
plt.show()
```



Data Pre-processing

In [ ]:
```
wine.isnull().sum()
```

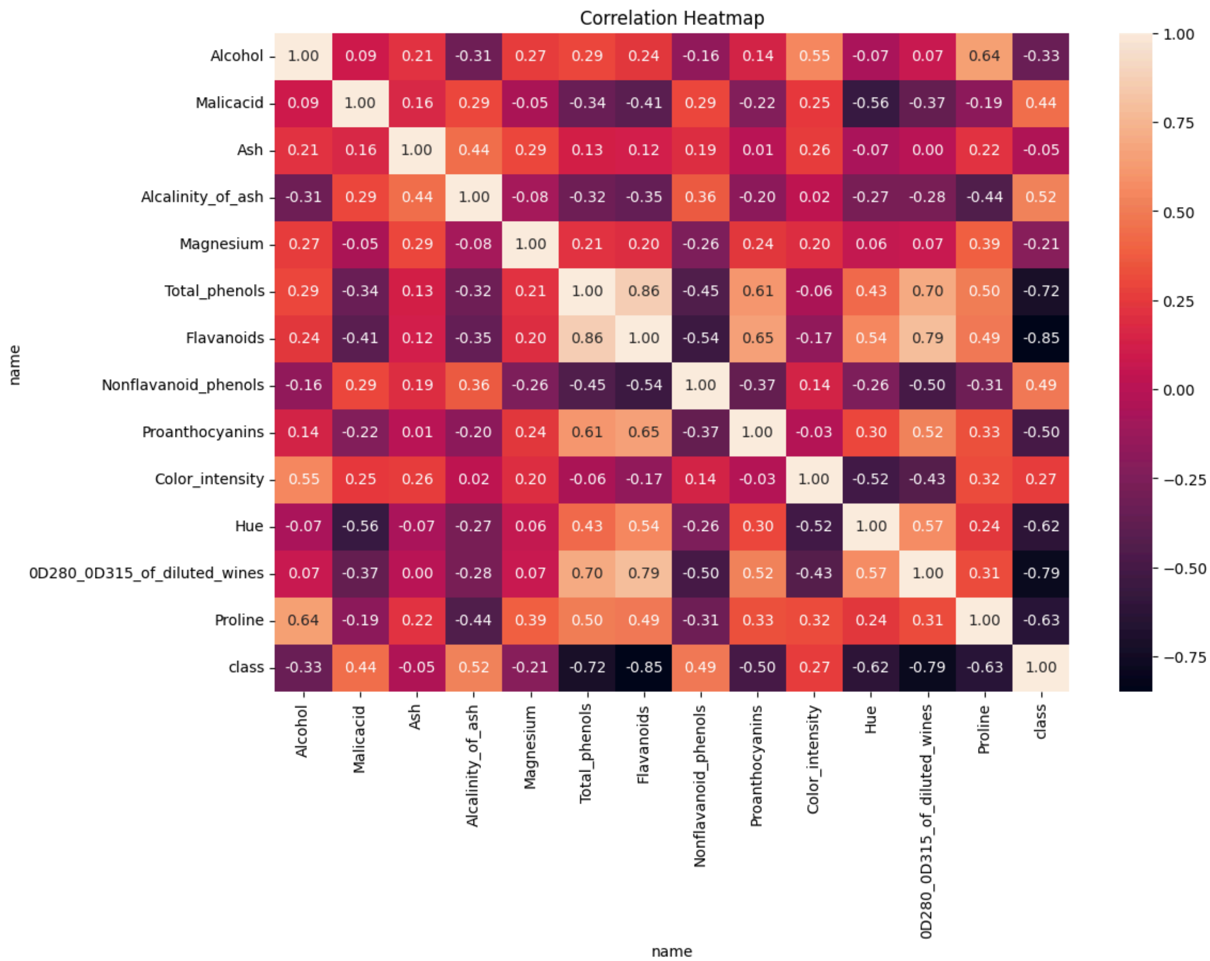
Out[ ]:
```
name
Alcohol                         0
Malicacid                       0
Ash                             0
Alcalinity_of_ash               0
Magnesium                       0
Total_phenols                   0
Flavanoids                      0
Nonflavanoid_phenols            0
Proanthocyanins                 0
Color_intensity                 0
Hue                             0
0D280_0D315_of_diluted_wines    0
Proline                         0
class                           0
dtype: int64
```

In [ ]:
```
scaler = StandardScaler()
wine_scaled = scaler.fit_transform(wine.drop('class', axis=1))
```

In [ ]:
```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
wine['class'] = le.fit_transform(wine['class'])
```

In [ ]:
```
plt.figure(figsize=(12, 8))
sns.heatmap(wine.corr(), annot=True, cmap='rocket', fmt='.2f')
```

```
plt.title('Correlation Heatmap')
plt.show()
```



Conclusion

Linear Regression is a fundamental technique in data analysis that is used to model the relationship between a dependent variable and one or more independent variables. The aim of linear regression is to find the best line that fits the data, which can be used to make predictions or forecasts. Moreover, Linear regression can be used for a variety of purposes, including predictive modeling, forecasting, exploratory data analysis, and model selection. It is a versatile technique that can be used for a variety of applications, including sales forecasting, stock price predictions, and even weather forecasting. Logistic Regression on the other hand can be use to find answers to questions that have two or more finite outcomes. We can also use logistic regression to pre-process data such as sorting data with a large range of values like bank transactions into a smaller and finite range of values.