Multivariate Linear Regression Practice

July 10, 2025

0.1 Miles Per Gallon

[2]:
```
#import Libraries
```

[5]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as pit
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import seaborn as sns
```

[8]:
```
#Load the Data sets

url = 'https://raw.githubusercontent.com/mwaskom/seaborn-data/master/raw/mpg.
 ↪csv'
df = pd.read_csv(url)
```

[9]:
```
#print the data set
df.head()
```

[9]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino

[10]:
```
#print summary
df.info()
```

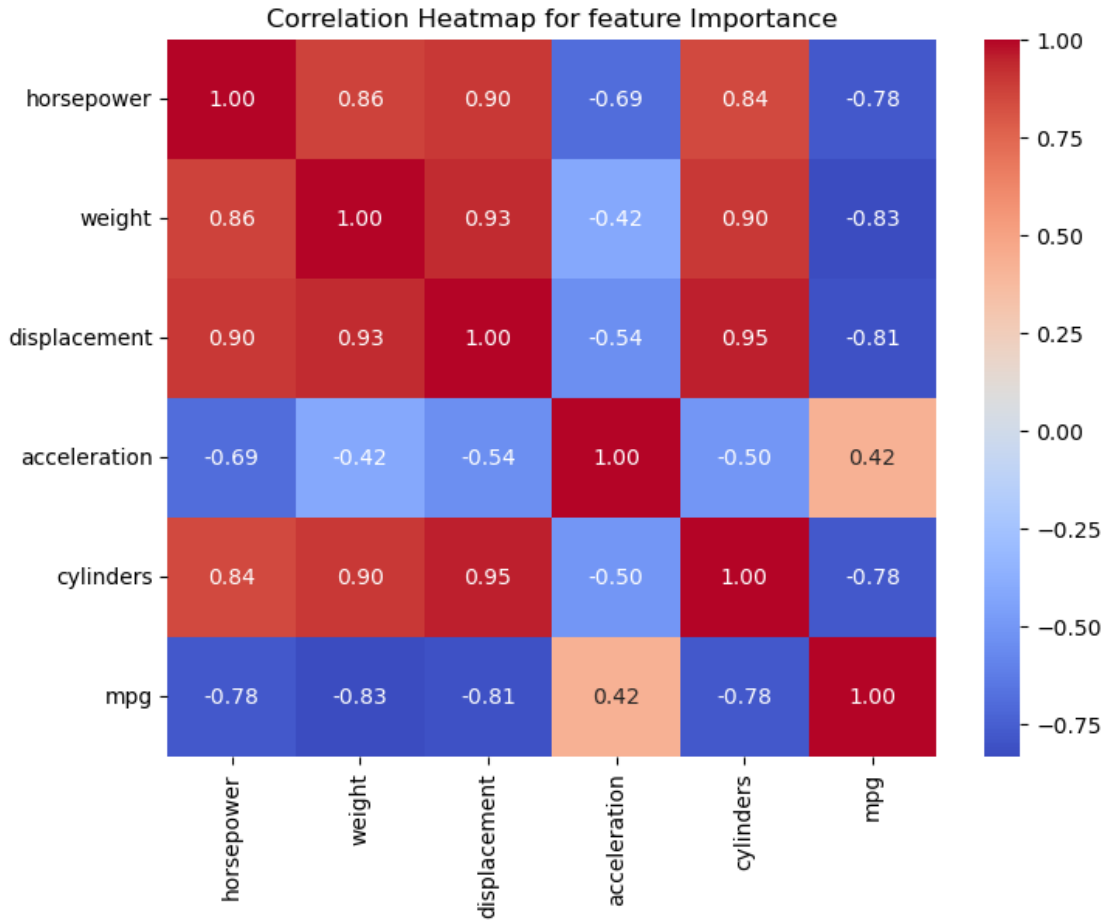
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    object
 4   weight        398 non-null    int64
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64
 7   origin        398 non-null    int64
 8   name          398 non-null    object
dtypes: float64(3), int64(4), object(2)
memory usage: 28.1+ KB
```

[13]:
```
df=df.dropna()
```

[36]:
```
features=['horsepower','weight','displacement','acceleration','cylinders']
target='mpg'
df['horsepower'] = pd.to_numeric(df['horsepower'], errors='coerce')
df = df.dropna()

#calculating corealtion matrix
correlation_matrix=df[features + [target]].corr()
```

[37]:
```
#plotting heatmap
```

[38]:
```
pit.figure(figsize=(8,6))
sns.heatmap(correlation_matrix,annot=True,cmap='coolwarm',fmt=".2f")
pit.title('Correlation Heatmap for feature Importance')
pit.show()
```



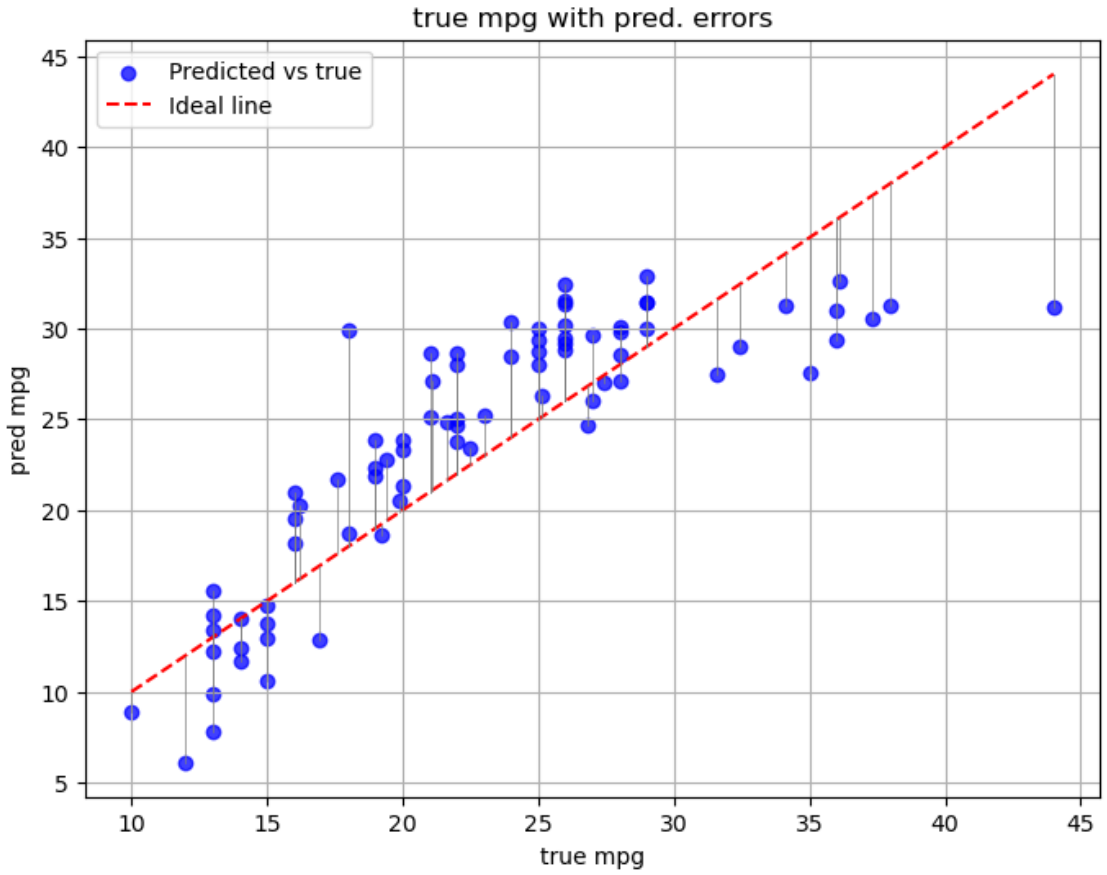
[39]:
```
X=df[['horsepower','weight','displacement','cylinders']].values
y=df['mpg'].values
```

[42]:
```
#splitting the data into train and test set
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.
  ↪2,random_state=42)

#creating a linear regression model
model=LinearRegression()

#train the model
model.fit(X_train,y_train)


#make the predictions on the test set
prediction=model.predict(X_test)

#evaluate the model
```

```
mse=mean_squared_error(y_test,prediction)
print(f"Mean Squared ErrorL { mse:.2f}")
```

```
Mean Squared ErrorL 17.89
```

[50]:
```
#rmse=4.22 off from actual value
pit.figure(figsize=(8,6))
pit.scatter(y_test,prediction,alpha=0.75,color='blue',label='Predicted vs true')
pit.
 ↪plot([min(y_test),max(y_test)],[min(y_test),max(y_test)],color='red',linestyle='--',label='
 ↪line')
#plotting prediction difference
for(t,p) in zip(y_test,prediction):
    pit.plot([t,t],[t,p],color='gray',linestyle='-',linewidth=0.5)

pit.xlabel("true mpg")
pit.ylabel("pred mpg")
pit.title("true mpg with pred. errors")
pit.legend()
pit.grid(True)
pit.show()
df.info()
```



```
<class 'pandas.core.frame.DataFrame'>
Index: 392 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           392 non-null    float64
 1   cylinders     392 non-null    int64
 2   displacement  392 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        392 non-null    int64
 5   acceleration  392 non-null    float64
 6   model_year    392 non-null    int64
 7   origin        392 non-null    int64
 8   name          392 non-null    object
dtypes: float64(4), int64(4), object(1)
memory usage: 30.6+ KB
```

[ ]: