

Simple Linear Regression

Using scikit-learn to implement simple Linear Regression. Check how Co2Emission is related to Co2Emissions. Create a model using train set, test using test set .

```
In [21]: import matplotlib.pyplot as plt
import pandas as pd
import pylab as pl
import numpy as np
%matplotlib inline
```

```
In [22]: df = pd.read_csv("FuelConsumptionCo2.csv")
df.head()
```

Out [22]:

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINE SIZE	CYLINDERS	TRANSMISSION	FUE
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5	
1	2014	ACURA	ILX	COMPACT	2.4	4	M6	
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	

Creating train and test dataset:

Train/Test Split dataset to mutually exclusive. We can use 80% of the entire data for training, and the 20% for testing. We create a mask using np.random.rand().

```
In [23]: msk = np.random.rand(len(df)) < 0.8
train = cdf[msk]
test = cdf[~msk]
```

Linear Regression

Training model using Training Set

```
In [24]: from sklearn import linear_model
regr = linear_model.LinearRegression()
train_x = np.asanyarray(train[['ENGINE_SIZE']])
train_y = np.asanyarray(train[['CO2EMISSIONS']])
regr.fit (train_x, train_y)

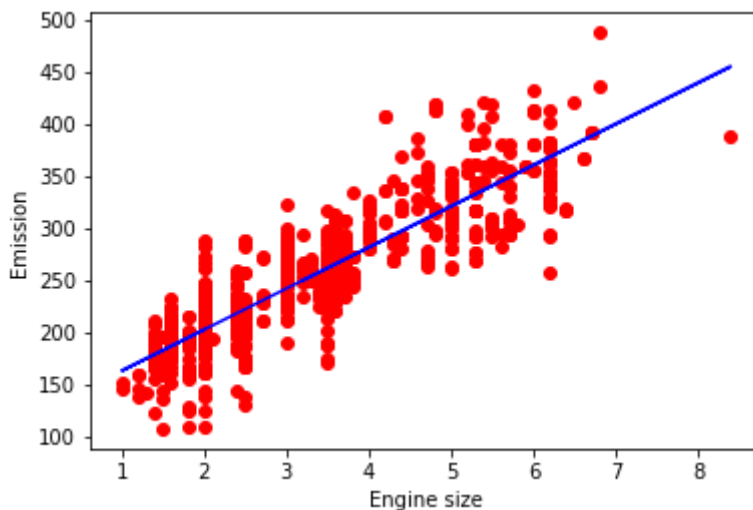
# The coefficients and intercept
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)
```

```
Coefficients:  [[39.44612413]]
Intercept:  [124.11232681]
```

Plotting to find how regression line fits:

```
In [28]: plt.scatter(train.ENGINE_SIZE, train.CO2EMISSIONS, color='red')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], '-b')
plt.xlabel("Engine size")
plt.ylabel("Emission")
```

```
Out[28]: Text(0, 0.5, 'Emission')
```



Evaluate using R2 score

```
In [29]: from sklearn.metrics import r2_score
test_x = np.asanyarray(test[['ENGINE_SIZE']])
test_y = np.asanyarray(test[['CO2EMISSIONS']])
test_y_hat = regr.predict(test_x)
```

```
In [30]: print("R2-score: %.2f" % r2_score(test_y_hat , test_y) )
```

```
R2-score: 0.70
```

In []: