ML0101EN-Reg-Polynomial-Regression-Co2-py-v1

November 14, 2018

#
Polynomial Regression

About this Notebook In this notebook, we learn how to use scikit-learn for Polynomial regression. We download a dataset that is related to fuel consumption and Carbon dioxide emission of cars. Then, we split our data into training and test sets, create a model using training set, evaluate our model using test set, and finally use model to predict unknown value.

0.0.1 Importing Needed packages

```
In [1]: import matplotlib.pyplot as plt
        import pandas as pd
        import pylab as pl
        import numpy as np
        %matplotlib inline
```

0.0.2 Downloading Data

To download the data, we will use !wget to download it from IBM Object Storage.

```
In [2]: !wget -O FuelConsumption.csv https://s3-api.us-geo.objectstorage.softlayer.net/cf-course
        print('Got it!')
```

```
--2018-11-14 15:28:16--  https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/Cogni
Resolving s3-api.us-geo.objectstorage.softlayer.net (s3-api.us-geo.objectstorage.softlayer.net).
Connecting to s3-api.us-geo.objectstorage.softlayer.net (s3-api.us-geo.objectstorage.softlayer.n
HTTP request sent, awaiting response... 200 OK
Length: 72629 (71K) [text/csv]
Saving to: FuelConsumption.csv

FuelConsumption.csv 100%[=====================>]  70.93K  --.-KB/s    in 0.04s

2018-11-14 15:28:16 (1.63 MB/s) - FuelConsumption.csv saved [72629/72629]

Got it!
```

Did you know? When it comes to Machine Learning, you will likely be working with large datasets. As a business, where can you host your data? IBM is offering a unique opportunity for businesses, with 10 Tb of IBM Cloud Object Storage: Sign up now for free

0.1 Understanding the Data

0.1.1 `FuelConsumption.csv`:

We have downloaded a fuel consumption dataset, `FuelConsumption.csv`, which contains model-specific fuel consumption ratings and estimated carbon dioxide emissions for new light-duty vehicles for retail sale in Canada. Dataset source

- **MODELYEAR** e.g. 2014
- **MAKE** e.g. Acura
- **MODEL** e.g. ILX
- **VEHICLE CLASS** e.g. SUV
- **ENGINE SIZE** e.g. 4.7
- **CYLINDERS** e.g 6
- **TRANSMISSION** e.g. A6
- **FUEL CONSUMPTION in CITY(L/100 km)** e.g. 9.9
- **FUEL CONSUMPTION in HWY (L/100 km)** e.g. 8.9
- **FUEL CONSUMPTION COMB (L/100 km)** e.g. 9.2
- **CO2 EMISSIONS (g/km)** e.g. 182 --> low --> 0

0.2 Reading the data in

```
In [3]: df = pd.read_csv("FuelConsumption.csv")

        # take a look at the dataset
        df.head()
```

```
Out[3]:    MODELYEAR   MAKE       MODEL VEHICLECLASS  ENGINESIZE  CYLINDERS  \
        0       2014  ACURA         ILX      COMPACT         2.0          4
        1       2014  ACURA         ILX      COMPACT         2.4          4
        2       2014  ACURA  ILX HYBRID      COMPACT         1.5          4
        3       2014  ACURA     MDX 4WD  SUV - SMALL         3.5          6
        4       2014  ACURA     RDX AWD  SUV - SMALL         3.5          6

          TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY  \
        0          AS5        Z                   9.9                  6.7
        1           M6        Z                  11.2                  7.7
        2          AV7        Z                   6.0                  5.8
        3          AS6        Z                  12.7                  9.1
        4          AS6        Z                  12.1                  8.7

           FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
        0                   8.5                        33           196
        1                   9.6                        29           221
        2                   5.9                        48           136
```

```
3                   11.1                          25            255
4                   10.6                          27            244
```

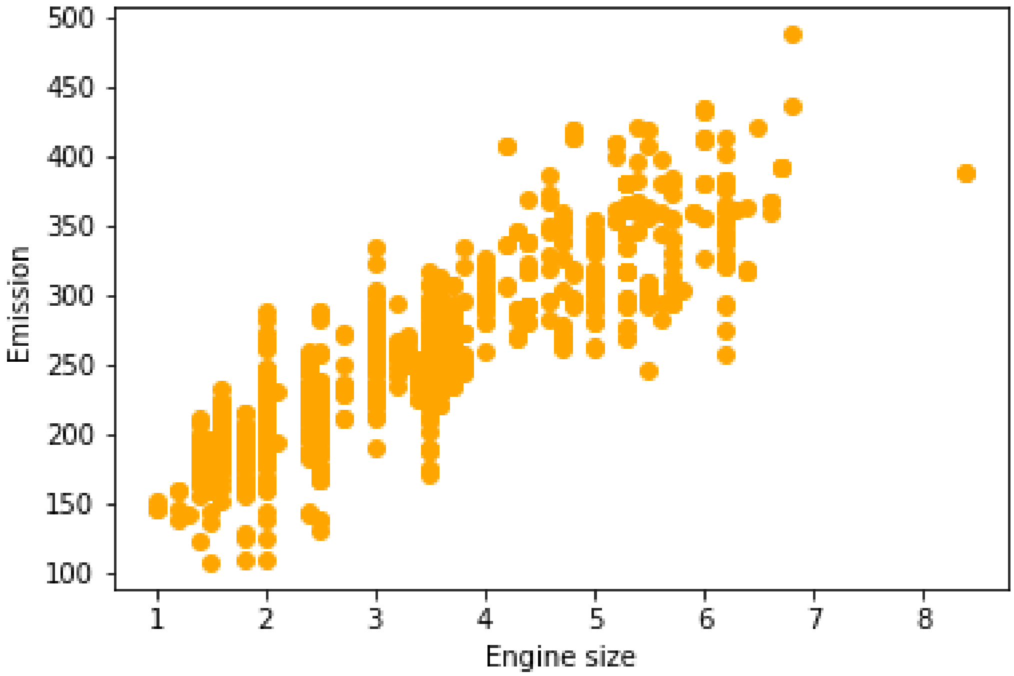
Lets select some features that we want to use for regression.

```
In [4]: cdf = df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']]
        cdf.head(9)
```

```
Out[4]:    ENGINESIZE  CYLINDERS  FUELCONSUMPTION_COMB  CO2EMISSIONS
        0         2.0          4                   8.5           196
        1         2.4          4                   9.6           221
        2         1.5          4                   5.9           136
        3         3.5          6                  11.1           255
        4         3.5          6                  10.6           244
        5         3.5          6                  10.0           230
        6         3.5          6                  10.1           232
        7         3.7          6                  11.1           255
        8         3.7          6                  11.6           267
```

Lets plot Emission values with respect to Engine size:

```
In [5]: plt.scatter(cdf.ENGINESIZE, cdf.CO2EMISSIONS,  color='orange')
        plt.xlabel("Engine size")
        plt.ylabel("Emission")
        plt.show()
```


Emission
Engine size