ML01-Reg-Simple-Linear-Regression-Co2-py-v1

May 4, 2019

#
Simple Linear Regression

About this Notebook In this notebook, we learn how to use scikit-learn to implement simple linear 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 your model using test set, and finally use model to predict unknown value

0.0.1 Importing Needed packages

```
In [4]: 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 [5]: !wget -O "./data/FuelConsumption.csv" https://s3-api.us-geo.objectstorage.softlayer.net

--2019-05-04 17:05:34--  https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/Cogn
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
HTTP request sent, awaiting response... 200 OK
Length: 72629 (71K) [text/csv]
Saving to: ./data/FuelConsumption.csv

./data/FuelConsumpt 100%[===================>]  70.93K  --.-KB/s    in 0.04s

2019-05-04 17:05:34 (1.81 MB/s) - ./data/FuelConsumption.csv saved [72629/72629]
```

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 [6]: df = pd.read_csv("./data/FuelConsumption.csv")

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

Out[6]:

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	FUELCONSUMPTION_COMB	FUELCONSUMPTION_COMB_MPG	CO2EMISSIONS
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	6.7	8.5	33	196
1	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7.7	9.6	29	221
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	5.8	5.9	48	136
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	9.1	11.1	25	255
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8.7	10.6	27	244

0.2.1 Data Exploration

Lets first have a descriptive exploration on our data.

```
In [7]: # summarize the data
        df.describe()
```

```
Out[7]:        MODELYEAR   ENGINESIZE    CYLINDERS  FUELCONSUMPTION_CITY  \
        count     1067.0  1067.000000  1067.000000           1067.000000
        mean      2014.0     3.346298     5.794752             13.296532
        std          0.0     1.415895     1.797447              4.101253
        min       2014.0     1.000000     3.000000              4.600000
        25%       2014.0     2.000000     4.000000             10.250000
        50%       2014.0     3.400000     6.000000             12.600000
        75%       2014.0     4.300000     8.000000             15.550000
        max       2014.0     8.400000    12.000000             30.200000

               FUELCONSUMPTION_HWY  FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  \
        count          1067.000000           1067.000000               1067.000000
        mean              9.474602             11.580881                 26.441425
        std               2.794510              3.485595                  7.468702
        min               4.900000              4.700000                 11.000000
        25%               7.500000              9.000000                 21.000000
        50%               8.800000             10.900000                 26.000000
        75%              10.850000             13.350000                 31.000000
        max              20.500000             25.800000                 60.000000

               CO2EMISSIONS
        count   1067.000000
        mean     256.228679
        std       63.372304
        min      108.000000
        25%      207.000000
        50%      251.000000
        75%      294.000000
        max      488.000000
```

Lets select some features to explore more.

```
In [8]: cdf = df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']]
        cdf.head(9)
```

```
Out[8]:    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
```