

COMPUTING SUBJECT: Machine Learning

TYPE: WORK ASSIGNMENT

IDENTIFICATION: Linear Regression Standard

COPYRIGHT: *Michael Claudius & Jens Peter Andersen*

DEGREE OF DIFFICULTY: Easy

TIME CONSUMPTION: 1 hour

EXTENT: < 60 lines

OBJECTIVE: Using a simple Dataframe
Using Scikit-Learn linear regression

COMMANDS:

IDENTIFICATION: LinerRegSimple/MICL

The Mission

Establishing a dataframe, which typically is the starting point for machine learning. Using Scikit-Learn's linear regression to find the regression line.

The problem

To find the best regression line for at training set of click data.

Useful links

<https://www.w3schools.com/Python/default.asp>

<https://docs.python.org/3/library/random.html>

https://www.tutorialspoint.com/python_data_structure/python_2darray.htm

Step 1: Establish a simple Dataframe

Start Jupyter Notebook and make a new notebook: LinearRegStandard

Import needed libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

Establish training set as a dataframe:

```
clickData = {'CostPerClick': [2.3, 2.1, 2.5, 4.5, 5.9, 4.1, 8.9],
             'TotalClicksPerDay': [89.0, 63.0, 71.0, 70.0, 80.0, 89.0, 150.0]}
trainingSet = pd.DataFrame(clickData)
trainingSet
```

Note datatype:

```
type(trainingSet)
```

Step 2: Establish features and labels

Extract labels:

```
clickLabels = np.c_[trainingSet["TotalClicksPerDay"]]
clickLabels
```

Note datatype:

```
type(clickLabels)
```

Extract feature:

```
clickFeatures = np.c_[trainingSet["CostPerClick"]]  
clickFeatures
```

Note datatype:

```
type(clickFeatures)
```

Step 3: Perform linear regression

Create linear regression model and fit it:

```
lin_reg = LinearRegression()  
lin_reg.fit(clickFeatures,clickLabels)
```

Step 4: Get slope and intersection

Get intersection:

```
intersection=lin_reg.intercept_[0]  
intersection
```

Get slope:

```
slope=lin_reg.coef_[0][0]  
slope
```

Step 5: Plot training set and regression line

Perform plotting:

```
plt.axis([0, 10, 0, 200])  
plt.scatter(clickFeatures, clickLabels)  
  
intersection=lin_reg.intercept_[0]  
slope=lin_reg.coef_[0]  
  
test_line = [(slope*item + intersection) for item in [0, 10]]  
plt.plot([0, 10], test_line)
```

Congratulations.