scientistmetrics

version 0.0.2

Duvérier DJIFACK ZEBAZE

Table des matières

1 powersetmodel function **1**

1.1 powersetmodel with linear regression 1

1.1.1 Datasets 1

1 powersetmodel function

powersetmodel is a function that return all subsets models giving a set a variables as features. This first version is based under linear regression and logistic regression.

1.1 powersetmodel with linear regression

1.1.1 Datasets

Considering Medical insurance costs dataset. This datasets was inspired by the book Machine Learning with R by @ . The data contains medical information and costs billed by health insurance companies. It contains 1338 rows of data and the following columns : age, gender, BMI, children, smoker, region and insurance charges.

```
# Load dataset
import pandas as pd
insurance = pd.read_csv("./donnee/insurance.csv",sep=",")
print(insurance.info())
```

```
## <class 'pandas.core.frame.DataFrame'>
## RangeIndex: 1338 entries, 0 to 1337
## Data columns (total 7 columns):
##  #   Column    Non-Null Count  Dtype
## ---  ------    --------------  -----
##  0   age       1338 non-null   int64
##  1   sex       1338 non-null   object
##  2   bmi       1338 non-null   float64
##  3   children  1338 non-null   int64
##  4   smoker    1338 non-null   object
##  5   region    1338 non-null   object
##  6   charges   1338 non-null   float64
## dtypes: float64(2), int64(2), object(3)
## memory usage: 73.3+ KB
## None
```

1.1.1.1 Columns description

- age : age of primary beneficiary
- sex : insurance contractor gender, female, male
- bmi : Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg/m^2) using the ratio of height to weight, ideally 18.5 to 24.9
- children : Number of children covered by health insurance / Number of dependents
- smoker : Smoking
- region : the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
- charges : Individual medical costs billed by health insurance

The dataset in available on https ://github.com/stedy/Machine-Learning-with-R-datasets.

Now, let load the powersetmodel function

```
# Load function
# Powerset
from scientistmetrics import powersetmodel
```

Let explains « charges » using other features

```
# Powerset model
ols_res = powersetmodel(DTrain=insurance,target="charges")
```

Check the length of the « ols_res » variables.

```
# Len of element
len(ols_res)
```

```
## 2
```

The function return two éléments :

- The first element is a list a all subsets train models
- The second element is a dataframe of result.

Summary of first model.

```
# First element
ols_model = ols_res[0]
ols_model[0].summary2()
```

```
## <class 'statsmodels.iolib.summary2.Summary'>
## """
##                   Results: Ordinary least squares
## ================================================================
## Model:              OLS              Adj. R-squared:     0.078
## Dependent Variable: charges          AIC:                20198.8425
```

```
## Date:               2023-08-25 00:26 BIC:                20208.5257
## No. Observations:   936              Log-Likelihood:     -10097.
## Df Model:           1                F-statistic:        79.67
## Df Residuals:       934              Prob (F-statistic): 2.30e-18
## R-squared:          0.079            Scale:              1.3761e+08
## ---------------------------------------------------------------------
##                 Coef.    Std.Err.   t    P>|t|    [0.025    0.975]
## ---------------------------------------------------------------------
## Intercept     3793.3197 1145.6379 3.3111 0.0010 1544.9971 6041.6423
## age            246.3784   27.6026 8.9259 0.0000  192.2082  300.5486
## ---------------------------------------------------------------------
## Omnibus:              283.356       Durbin-Watson:          1.994
## Prob(Omnibus):        0.000         Jarque-Bera (JB):       612.786
## Skew:                 1.727         Prob(JB):               0.000
## Kurtosis:             4.944         Condition No.:          124
## =====================================================================
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors
## is correctly specified.
## """
```

The second datasets

```
# Second elemnt
ols_metrics = ols_res[1]
print(ols_metrics.columns)
```

```
## Index(['predictor', 'count', 'aic', 'bic', 'rsquared', 'adj. rsquared',
##        'expl. var. score', 'max error', 'mean abs. error', 'mean sq. error',
##        'median abs. error', 'r2 score', 'mean abs. percentage error',
##        'likelihood test ratio'],
##       dtype='object')
```

```
print(ols_metrics)
```

```
##                                predictor  ...  likelihood test ratio
## 0                                    age  ...            1137.252767
## 1                                    sex  ...            1210.243862
## 2                                    bmi  ...            1175.417401
## 3                               children  ...            1208.071940
## 4                                 smoker  ...             359.954462
## ..                                   ...  ...                    ...
## 58           smoker+bmi+sex+age+region  ...               7.099104
## 59      smoker+children+sex+age+region  ...              92.183682
## 60      smoker+children+bmi+age+region  ...               0.000239
## 61      smoker+children+bmi+sex+region  ...             257.160460
## 62  smoker+children+bmi+sex+age+region  ...               0.000000
##
## [63 rows x 14 columns]
```