Universidad **Ricardo Palma**

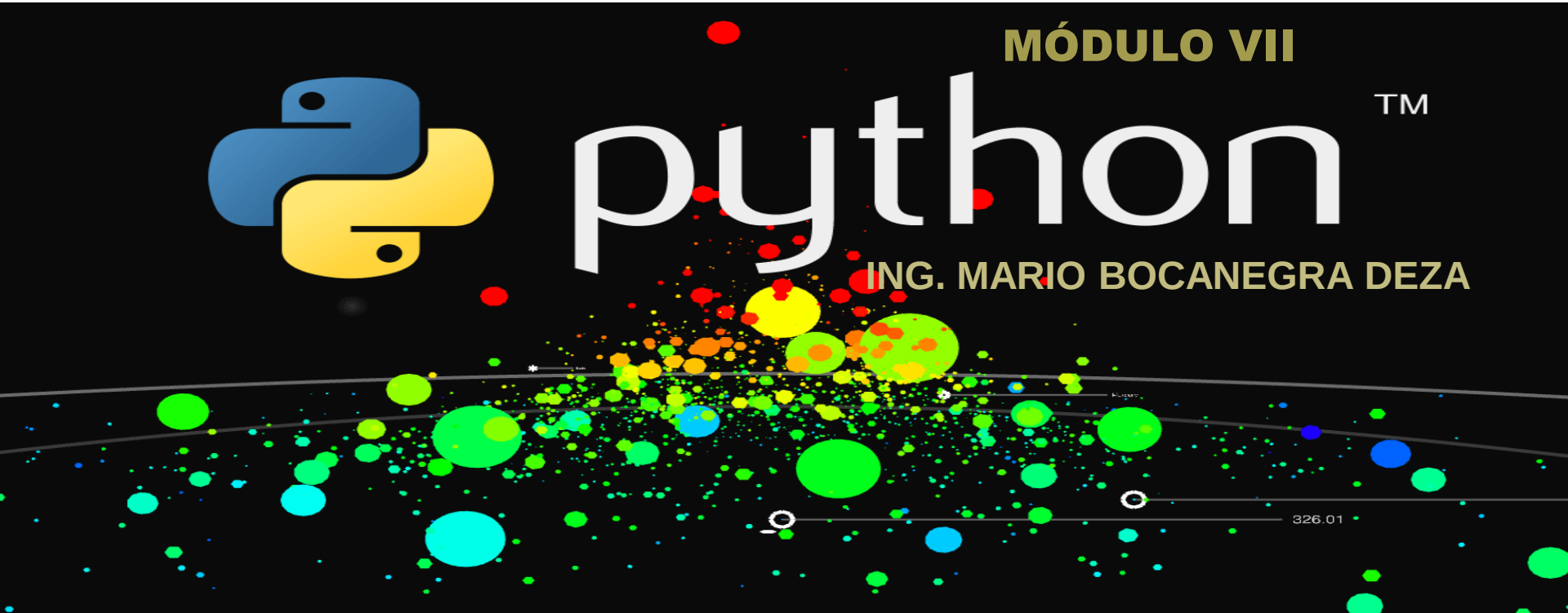
RECTORADO

Formamos seres humanos para una cultura de paz

Primer Programa de Especialización

INTRODUCCIÓN AL DATA SCIENCE

REGRESIÓN LINEAL CON PYTHON

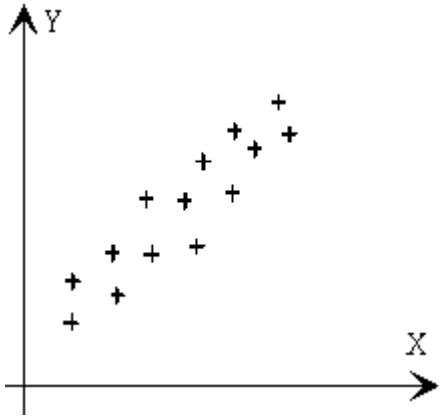


Contenido

- ✓Regresión Lineal Ideas Generales
- ✓ Principales librerías de Regresión Lineal

Regresión Lineal

- ✓ Uno de los aspectos más relevante de la Estadística es el análisis de la relación o dependencia entre variables.
- ✓ Resulta de interés conocer el efecto que una variable o varias variables pueden causar sobre otra e incluso predecir en mayor o menor grado valores en una variable a partir de otra.



Regresión lineal simple Y = f(x)

Regresión lineal múltiple Y = f(x, w, z).

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

DataSet (salario.csv)

```
In [44]: import pandas as pd
         import matplotlib.pyplot as plt
         %matplotlib inline
         from pandas.tools.plotting import scatter_matrix
         from scipy import stats
         import statsmodels.api as sm
         from sklearn.linear_model import LinearRegression
```

```
In [2]: df=pd.read_csv("C:/Users/Mario/Downloads/salario.csv")
```

	rank	discipline	yrs.since.phd	yrs.service	sex	salary
0	Prof	B	19	18	Male	139750
1	Prof	B	20	16	Male	173200
2	AsstProf	B	4	3	Male	79750
3	Prof	B	45	39	Male	115000
4	Prof	B	40	41	Male	141500
5	AssocProf	B	6	6	Male	97000
6	Prof	B	30	23	Male	175000
7	Prof	B	45	45	Male	147765
8	Prof	B	21	20	Male	119250
9	Prof	B	18	18	Female	129000

Análisis descriptivo

```
In [4]: df[['salary','yrs.since.phd','yrs.service']].describe()
```

Out[4]:

	salary	yrs.since.phd	yrs.service
count	397.000000	397.000000	397.000000
mean	113706.458438	22.314861	17.614610
std	30289.038695	12.887003	13.006024
min	57800.000000	1.000000	0.000000
25%	91000.000000	12.000000	7.000000
50%	107300.000000	21.000000	16.000000
75%	134185.000000	32.000000	27.000000
max	231545.000000	56.000000	60.000000

Función Correlación

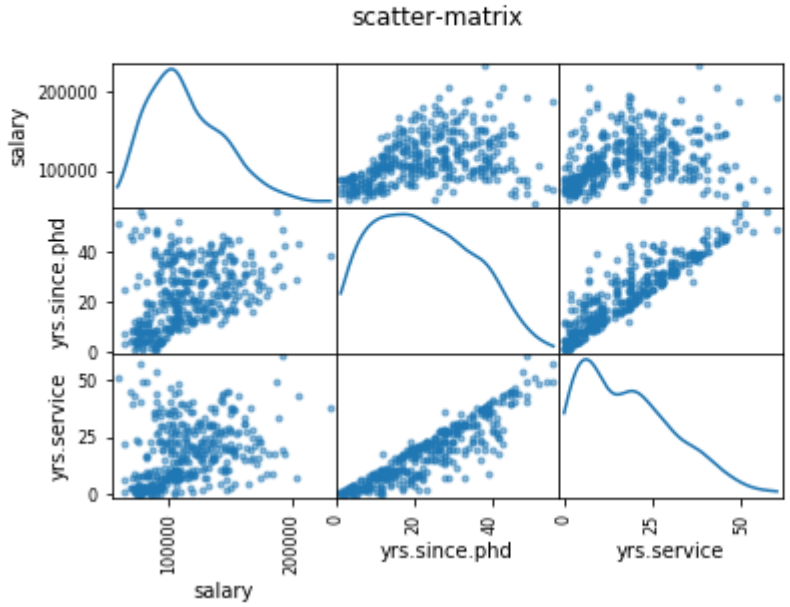
```
In [7]: df[['salary','yrs.since.phd','yrs.service']].corr()
```

Out[7]:

	salary	yrs.since.phd	yrs.service
salary	1.000000	0.419231	0.334745
yrs.since.phd	0.419231	1.000000	0.909649
yrs.service	0.334745	0.909649	1.000000

Gráficos de Correlación

```
In [8]: scatter_matrix(df[['salary','yrs.since.phd','yrs.service']], alpha=0.5,diagonal='kde')
        plt.suptitle('scatter-matrix')
        plt.show()
```



Librería scipy.stats.linregress

Es seguro | https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html

SciPy.org Sponsored By ENTHOUGHT

Scipy.org | Docs | SciPy v0.19.0 Reference Guide | Statistical functions (scipy.stats) | inde

scipy.stats.linregress

scipy.stats.linregress(*x*, *y=None*) [sou

Calculate a linear least-squares regression for two sets of measurements.

Parameters: **x, y** : *array_like*
Two sets of measurements. Both arrays should have the same length. If only x is given (and y=None), then it must be a two-dimensional array where one dimension has length 2. The two sets of measurements are then found by splitting the array along the length-2 dimension.

Returns: **slope** : *float*
slope of the regression line
intercept : *float*
intercept of the regression line
rvalue : *float*
correlation coefficient
pvalue : *float*
two-sided p-value for a hypothesis test whose null hypothesis is that the slope is zero.
stderr : *float*
Standard error of the estimated gradient.

Librería scipy.stats.linregress

```
In [57]: slope, intercept, r_value, p_value, std_err = stats.linregress(df['yrs.since.phd'],df['salary'])

In [34]: print(intercept,slope,std_err,r_value,p_value)
         91718.6854469 985.342124121 107.365125554 0.419231106803 2.49504231391e-18
```

Librería statsmodels

statsmodels.sourceforge.net/stable/regression.html

StatsModels
Statistics in Python

Install | Support | Bugs | Develop | Examples | FAQ | previous | next |

Table Of Contents

- Linear Regression
  - Examples
  - Technical Documentation
    - References
    - Attributes
  - Module Reference
    - Model Classes
    - Results Classes

Previous topic

tatsmodels.stats.contrast.Contrast
esults.summary_frame

Next topic

rdinary Least Squares

This Page

Show Source

Linear Regression

Linear models with independently and identically distributed errors, and for errors with heteroscedasticity or autocorrelation. This module allows estimation by ordinary least squares (OLS), weighted least squares (WLS), generalized least squares (GLS), and feasible generalized least squares with autocorrelated AR(p) errors.

See Module Reference for commands and arguments.

Examples

```
# Load modules and data
import numpy as np
import statsmodels.api as sm
spector_data = sm.datasets.spector.load()
spector_data.exog = sm.add_constant(spector_data.exog, prepend=False)

# Fit and summarize OLS model
mod = sm.OLS(spector_data.endog, spector_data.exog)
res = mod.fit()
print res.summary()
```

Librería statsmodels

```
In [52]: results = sm.OLS(df['salary'],sm.add_constant(df[['yrs.since.phd','yrs.service']])).fit()

In [54]: print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 salary   R-squared:                       0.188
Model:                            OLS   Adj. R-squared:                  0.184
Method:                 Least Squares   F-statistic:                     45.71
Date:                Fri, 14 Apr 2017   Prob (F-statistic):           1.40e-18
Time:                        22:10:39   Log-Likelihood:                -4617.9
No. Observations:                 397   AIC:                             9242.
Df Residuals:                     394   BIC:                             9254.
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [95.0% Conf. Int.]
--------------------------------------------------------------------------------
const          8.991e+04   2843.560     31.620      0.000      8.43e+04  9.55e+04
yrs.since.phd  1562.8889    256.820      6.086      0.000      1057.981  2067.797
yrs.service    -629.1014    254.469     -2.472      0.014     -1129.389  -128.814
==============================================================================
Omnibus:                       14.927   Durbin-Watson:                   1.867
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               15.947
Skew:                           0.429   Prob(JB):                     0.000344
Kurtosis:                       3.478   Cond. No.                         69.6
==============================================================================
```

Librería sklearn

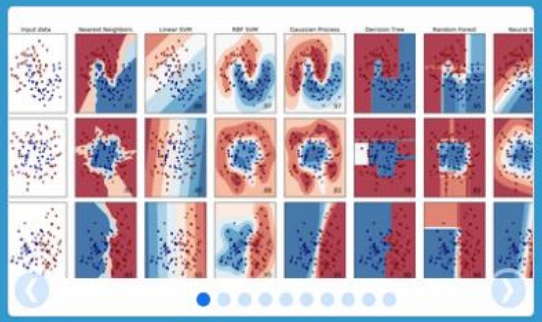
scikit-learn.org/stable/index.html

scikit learn

Home Installation Documentation Examples

Google Custom Search Search

powered by Google



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

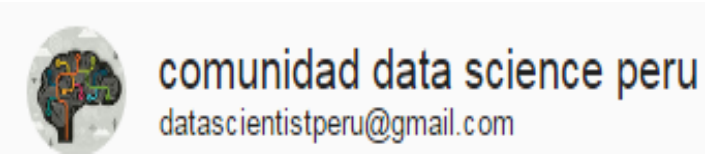
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Librería sklearn

```
In [58]: lr = LinearRegression()
         lr.fit(df[['yrs.since.phd','yrs.service']], df['salary'])

Out[58]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)

In [59]: print(lr.intercept_,lr.coef_)

         89912.1844638 [ 1562.88890188  -629.10138909]
```


fanpage
facebook


Comunidad
Data Science
Perú
@comunidaddatascience


comunidad data science peru
datascientistperu@gmail.com