Machine Learning

Simple & Multiple Linear Regression

1. Simple Linear Regression

Data handling!

1. Defining the business problem.

2. Defining the data set.

3. Exploratory analysis.

4. Prove the relationship between the two variables using:

- Standard deviation
- Variance
- Correlation

5. Observe the Correlation using the Pearson function, for example:
(from scipy.stats.stats import pearsonr)
(+1, -1 or 0)

6. Check the relationship with a graph

7. It is not possible to create a Regression Model if there is no relationship between the variables!

8. To build the Linear Regression Model (in Python) we can use:

- StatsModels package
- Scikit-Learn package

What is learned!

9. The result of the Regression Model are the parameters (**coefficients** a & b), with which we set up the Regression formula.

10. With the model trained, offer new data for him to make predictions.

1.1 Details - check which data format the algorithm expects to receive

StatsModel

Requires variable X to be a matrix.

We can add a constant to the value of X, generating a matrix

Scikit-Learn

Can use the RESHAPE function to change the data format.

X must be a matrix and Y can be a vector.

Format: numpy.ndarray

Predictions

The value of the input data must also be in the form of a matrix!

Cost Function of a Regression Model

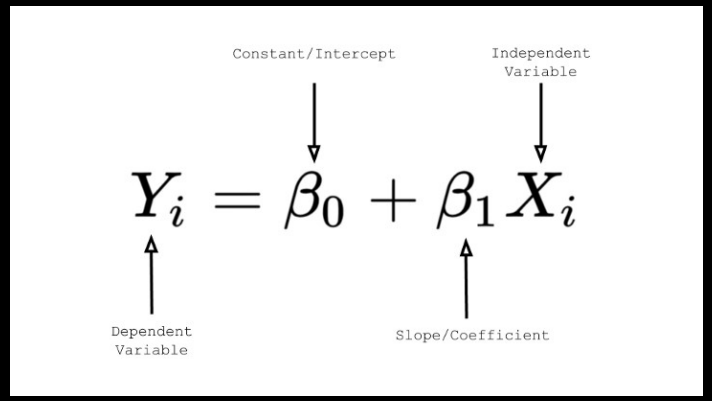
The objective of linear regression is to search for the equation of a regression line that minimizes the sum of the squared errors, the difference between the observed value of y and the predicted value. There are some methods for minimizing Cost Function such as:

- Pseudo-inversion
- Factoring
- Gradient Descent (more used!)

1.2 Cost Function of a Linear Regression Model

- We use the squared error in the cost function which will be minimized by the downward gradient.

- A cost function calculates the error rate of the model, and our goal is to minimize this function, to have the least possible error and thus increase the accuracy of the predictive model.

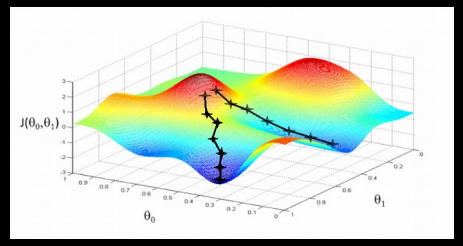


$$\frac{1}{2n} * \sum (h(X) - y)^2$$

Cost Function

$$w = (X^T X)^{-1} X^T y$$

Pseudo-inversion



$$J(w) = \frac{1}{2n} \sum (Xw - y)^2$$

$$w_j = w_j - \alpha * \frac{\partial}{\partial w} J(w)$$

Gradient Descent

$y_i - f(x_i)$

$f(x_i) - y_i$

$| f(x_i) - y_i |$

$(y_i - f(x_i))^2$

$$\text{Mean absolute error (MAE)} = \sum_{i=1}^{n} |f(x_i) - y_i|$$

$$\text{SSE/MSE} = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} (y_i - f(x_i))^2}$$

Total de Vendas (R$)	Total de Vendas Previsto (R$)
1245900	1278450
1302763	1334789
1345119	1320876

Variável Resposta (y)

Previsão f(x)

2. Multiple Linear Regression

1. Interpreting simple and multiple linear regression models:

- **F Test for Global Significance** *(look at the final result of the model)*
- **Individual Significance Tests** *(are tests for each of the explanatory variables)*
- **R2 Coefficients and Adjusted R2** *(coefficients used to compare different models)*
- **Coefficients** *(result of the training process)*

General Rules

Model is useful to predict the price, if the p-value of the F test is less than 0.05

There is evidence that a variable is related to the predicted value, if the p-value is less than 0.05

The objective of the regression is to find the coefficients that make it possible to construct the regression equation and make the predictions.

R2 indicates how much of the variability of y is explained by the predictor variables. It may be necessary to include more variables in the model to increase this coefficient.

2.1 Interpreting the p-value

The p-value is widely used to interpret regression models or even when we use statistical analysis.

P = 0.001: very unlikely to be by random.
P = 0.05: unlikely to be by random.
P = 0.5: likely to be by random.
P = 0.75: very likely to be by random.

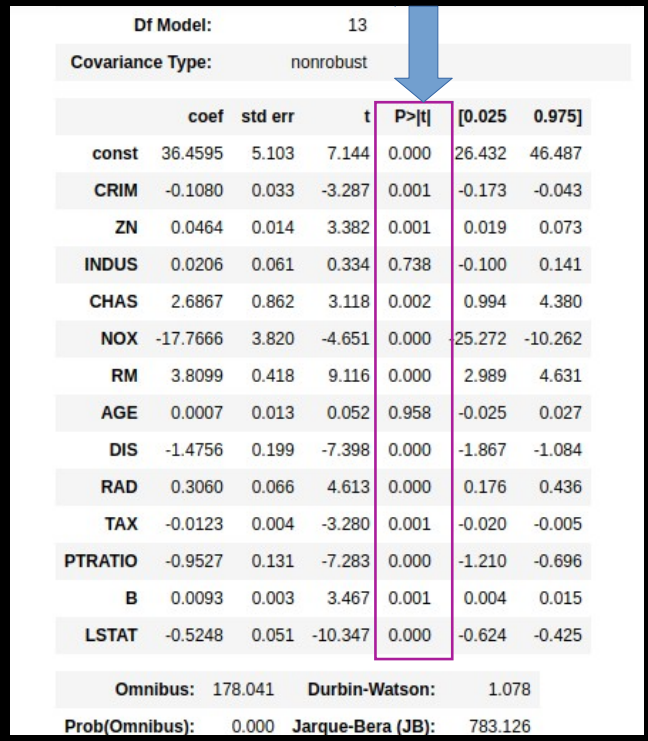
Statistical point of view

The p-value is used to actually assess the strength of the null and alternative hypotheses.

Point of view in data science

A **small p-value** can be inferred that **there is an association** between the predictor and the response, this means that we reject the null hypothesis.
A large p-value can be deduced that there is no association between the predictor and the response.

Example:

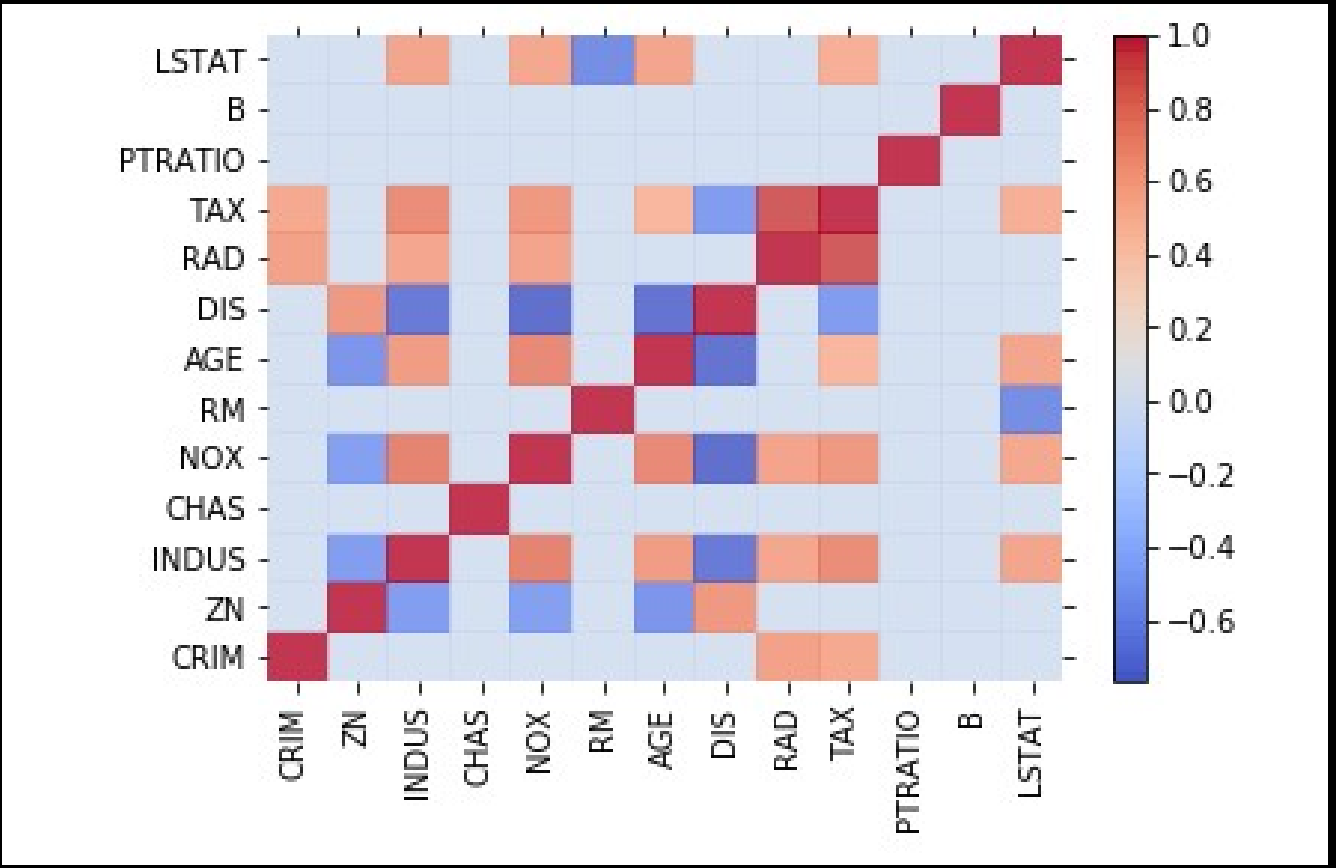


Df Model:			13			
Covariance Type:			nonrobust			
	coef	std err	t	P>|t|	[0.025	0.975]
const	36.4595	5.103	7.144	0.000	26.432	46.487
CRIM	-0.1080	0.033	-3.287	0.001	-0.173	-0.043
ZN	0.0464	0.014	3.382	0.001	0.019	0.073
INDUS	0.0206	0.061	0.334	0.738	-0.100	0.141
CHAS	2.6867	0.862	3.118	0.002	0.994	4.380
NOX	-17.7666	3.820	-4.651	0.000	-25.272	-10.262
RM	3.8099	0.418	9.116	0.000	2.989	4.631
AGE	0.0007	0.013	0.052	0.958	-0.025	0.027
DIS	-1.4756	0.199	-7.398	0.000	-1.867	-1.084
RAD	0.3060	0.066	4.613	0.000	0.176	0.436
TAX	-0.0123	0.004	-3.280	0.001	-0.020	-0.005
PTRATIO	-0.9527	0.131	-7.283	0.000	-1.210	-0.696
B	0.0093	0.003	3.467	0.001	0.004	0.015
LSTAT	-0.5248	0.051	-10.347	0.000	-0.624	-0.425

Omnibus:	178.041	Durbin-Watson:	1.078
Prob(Omnibus):	0.000	Jarque-Bera (JB):	783.126

2.2 Interpreting the Correlation Matrix

+1: strong positive correlation.
0: there is no correlation.
-1: strong negative correlation.



2.3 Multicollinearity

1. Objective in Machine Learning:

To make the model as **generalizable** as possible so that when it presents new values it can make predictions.

2. Identify the problem of Multicollinearity between explanatory variables, if you have two variables with the same type of information the model will be biased, the ideal is to remove one of the variables so that the model is more generalizable.

3. To identify Multicollinearity we use Eigenvalues and Eigenvectors.

Eigenvectors are a way to recombine the variance between variables, creating new resources by accumulating all shared variance.

Such a recombination can be obtained using the NumPy function **linalg.eig**, resulting in a vector of eigenvalues (representing the amount of recombined variance for each new variable) and eigenvectors (a matrix telling us how the new variables relate to the old ones).

2.4 Gradient Descent

1. Detail for Multiple Linear Regression
(the data needs to be on the same scale)

Feature Scaling

- We can apply Feature Scaling through Standardization or Normalization.

- Normalization scales data with intervals between 0 and 1.

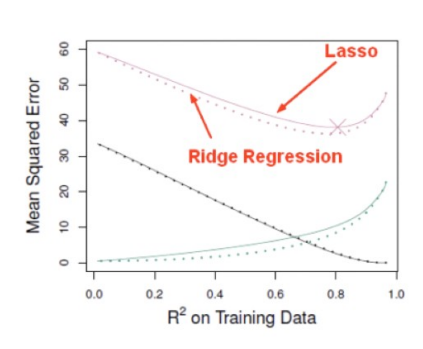
- Standardization divides the mean by the standard deviation to obtain a unit of variance.

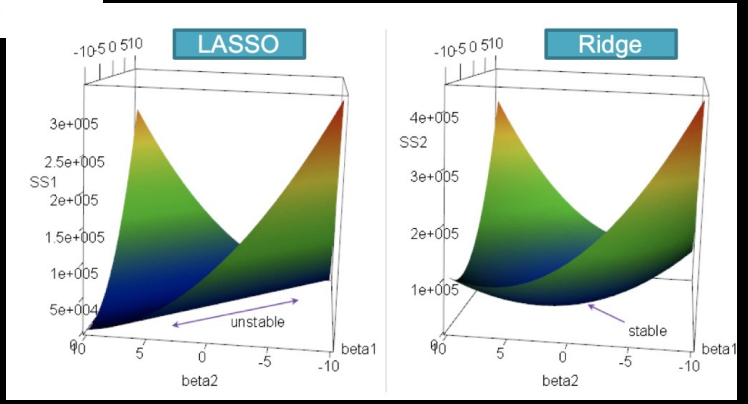
NOTE:
When standardization is done, to interpret the results it is necessary to “de-standardize” the data, so we will have the real values of the coefficients!

2.5 Regularization

1. There are 3 methods that help us when the number of variables is very large:

- Selection of a subset of coefficients
- Reduce the dimension
- Reduce the value of the coefficients (**Regularization**)





Shrinkage Methods

Ridge Regression

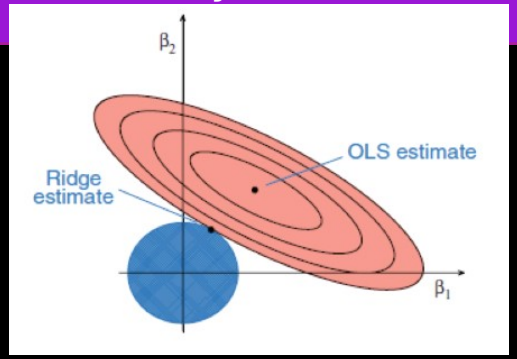
LASSO Regression
(Least Absolute Shrinkage and Selection Operator)

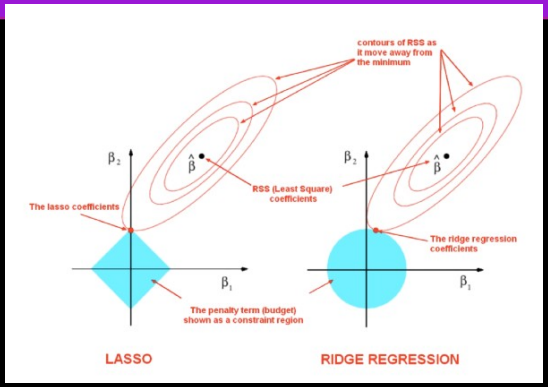
$$\hat{\beta}^{ridge} = \underset{\beta \in \mathbb{R}}{argmin} \|y - XB\|_2^2 + \lambda \|B\|_2^2$$

$$\sum_{i=1}^{n} (y_i - \sum_j x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Ridge Regression is a model regularization method whose main objective is to smooth attributes that are related to each other and that increase the noise in the model (multicollinearity).

LASSO has the same mechanism for penalizing the coefficients with a high degree of correlation with each other, but it uses the mechanism for penalizing the coefficients according to their absolute value.





3. Logistic Regression - **Classification**

1. A Logistic Regression is a supervised learning **Classification** algorithm.

In Logistic Regression, the response variable is binary:

1 - event of interest (success)
2 - complementary event (failure)

$$g(x) = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right)$$

$$g(x) = ln\left(\frac{\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}}{1-\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}}\right) = ln\left(\frac{\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}}{\frac{1}{1+e^{\beta_0+\beta_1 x}}}\right)$$

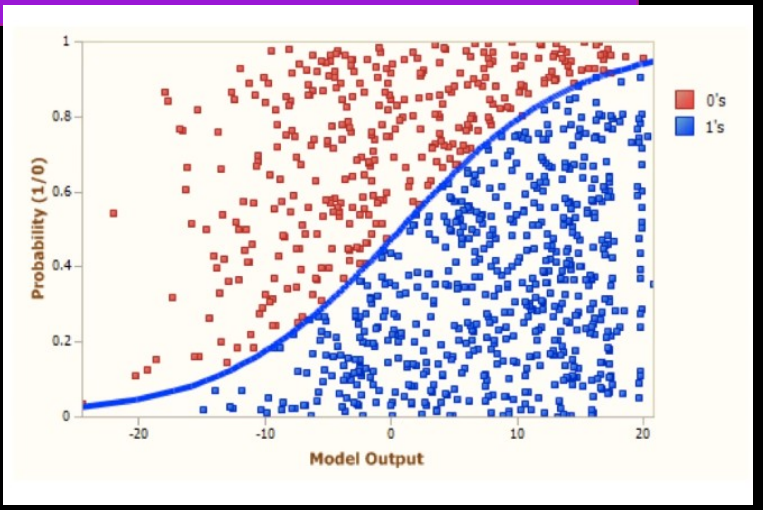
Transformação logit

$$g(x) = ln\left(e^{\beta_0+\beta_1 x}\right) = \beta_0 + \beta_1 x$$

↑ Logaritmo

2. Logistic regression is a statistical technique that aims to model, from a set of observations, the "logistical" relationship between a **response (categorical) variable** and a series of numerical explanatory variables (**continuous, discrete) and / or categorical.**

Logistic Regression is useful for modeling the **probability** that an event will occur as a function of other factors. It is a **generalized linear model** that uses the **LOGIT** function as a link function.



Logistic Regression is widely used in Medicine, Insurance Area, Financial Institutions, Econometrics.