Iliya Valchanov
Ivan Manov

Machine Learning in Python

Course Notes

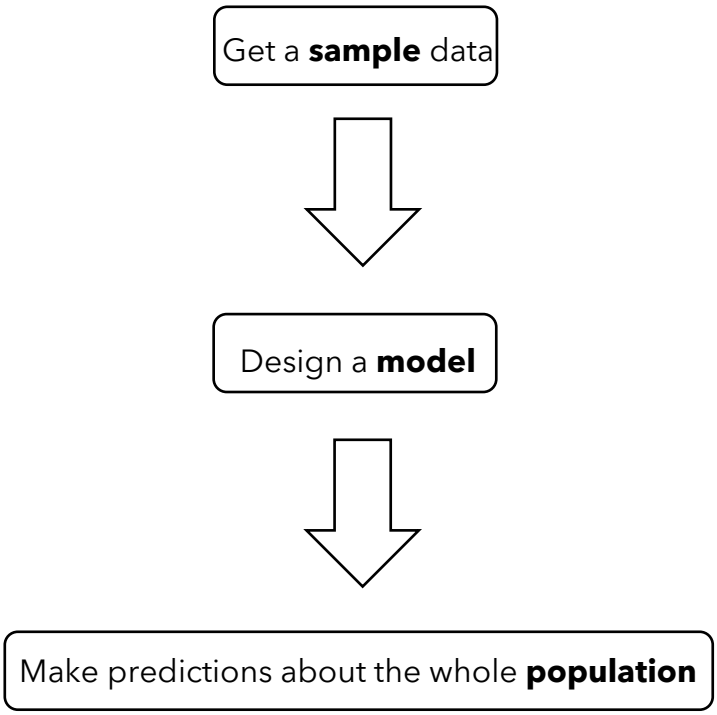
365 DataScience

Table of Contents

Abstract

Regression analysis is one of the most widely used methods for predictions, applied whenever we have a causal relationship between variables. A large portion of the predictive modelling that occurs in practice is carried out through regression analysis. It becomes extremely powerful when combined with techniques like factor analysis.

Regression models help us make predictions about the population based on sample data.



Section 1: Linear Regression

A linear regression is a linear approximation of a causal relationship between two or more variables. It is probably the most fundamental machine learning method and a starting point for the advanced analytical learning path of every aspiring data scientist.

1.1 The linear regression model

As many other statistical techniques, regression models help us make predictions about the population based on sample data.

Variables:

Dependent (predicted): Y

Independent (predictors): x_1, x_2, $x_3 \ldots x_k$

Y is a function of the ***x*** variables, and the regression model is a linear approximation of this function. The equation describing how Y is related to *x* is:

Simple linear regression model (population):

$$Y = \beta_0 + \beta_1 x_1 + \varepsilon$$

Y – dependent variable

x_1 – independent variable

β_0– constant/intercept

β_1– coefficient – quantifies the effect of x_1 on $\hat{y}$

ε – error of estimation

Simple linear regression equation (sample):

$$\hat{y} = b_0 + b_1 x_1$$

$\hat{y}$ – estimated/predicted value

b_0– coefficient – estimate of β_0

b_1– coefficient – estimate of β_1

x_1 – sample data for the independent variable

ε – error of estimation

When using regression analysis, the goal is to predict the value of Y, based on the value of x.

1.2 Correlation vs regression

Correlation - measures the degree of relationship between two variables

Regression - shows how one variable affects another or what changes it causes to the other

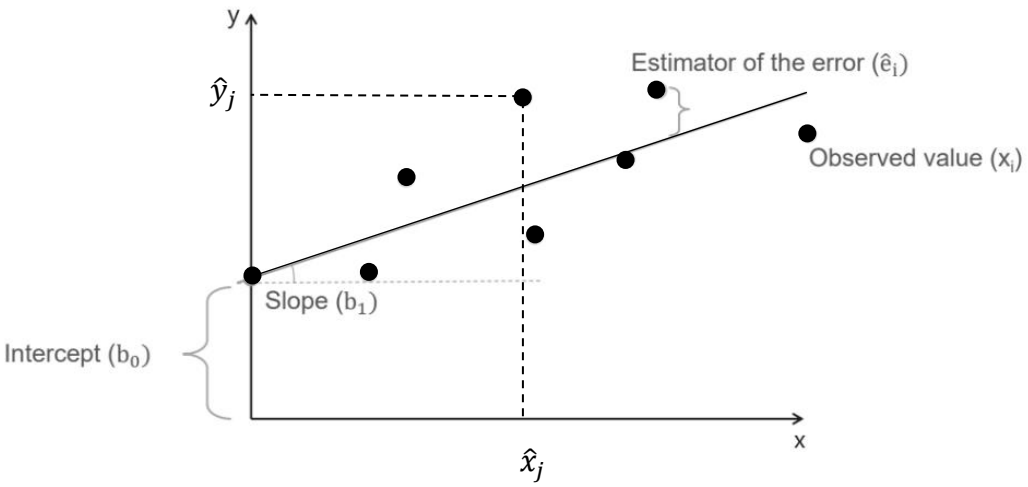
Correlation	Regression
The relationship between two variables	How one variable affects another
Movement together	Cause and effect
Symmetrical	One-way
A single point	A line

Comparison between correlation and regression

Linear regression analysis is known for the best fitting line that goes through the data points and minimizes the distance between them.

1.3 Geometrical representation of the Linear Regression Model

$$\hat{y} = b_0 + b_1 x_1$$



The linear regression model

Data points – the observed values (x)

b_0 - a constant - the intercept of the regression line with the y axis

b_1 - the slope of the regression line – shows how much y changes for each unit change of x

$\hat{e}_i$ – estimator of the error - the distance between the observed values and the regression line

$\hat{y}$ - the value predicted by the regression line

Regression line – the best fitting line through the data points

1.4 Regression in Python

Coding steps:

- Importing the relevant libraries
- Loading the data
- Defining the dependent and the independent variables following the regression equation
- Exploring the data
    - Plotting a scatter plot
- Regression itself
    - Adding a constant
    - Fitting the model according to the OLS method with a dependent variable *y* and an independent variable *x*
        - Ordinary least squares (OLS) – a method for finding the line which minimizes the SSE
    - Printing a summary of the regression

OLS Regression Results

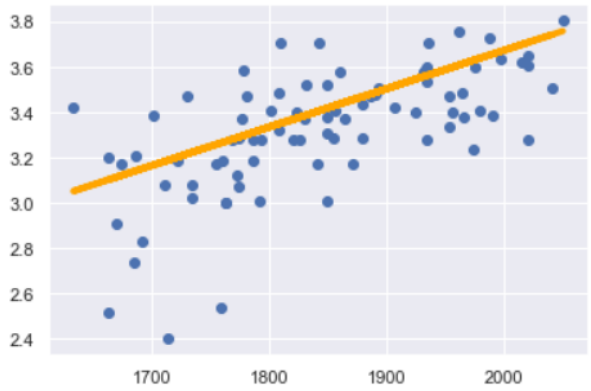
Dep. Variable:	GPA	**R-squared:**	0.406
Model:	OLS	**Adj. R-squared:**	0.399
Method:	Least Squares	**F-statistic:**	56.05
Date:	Tue, 08 Feb 2022	**Prob (F-statistic):**	7.20e-11
Time:	16:31:00	**Log-Likelihood:**	12.672
No. Observations:	84	**AIC:**	-21.34
Df Residuals:	82	**BIC:**	-16.48
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.2750	0.409	0.673	0.503	-0.538	1.088
SAT	0.0017	0.000	7.487	0.000	0.001	0.002

Omnibus:	12.839	**Durbin-Watson:**	0.950
Prob(Omnibus):	0.002	**Jarque-Bera (JB):**	16.155
Skew:	-0.722	**Prob(JB):**	0.000310
Kurtosis:	4.590	**Cond. No.**	3.29e+04

A summary table in Jupyter Notebook

- Creating a scatter plot
- Defining the regression equation
- Plotting the regression line



Plotted regression line in Jupyter Notebook

1.5 Interpreting the regression table

Summary table and important regression metrics

- ***A model summary***

Dep. Variable:	GPA	**R-squared:**	0.406
Model:	OLS	**Adj. R-squared:**	0.399
Method:	Least Squares	**F-statistic:**	56.05
Date:	Tue, 08 Feb 2022	**Prob (F-statistic):**	7.20e-11
Time:	16:36:35	**Log-Likelihood:**	12.672
No. Observations:	84	**AIC:**	-21.34
Df Residuals:	82	**BIC:**	-16.48
Df Model:	1		
Covariance Type:	nonrobust		

Model summary table

Dep. Variable - the dependent variable, *y*; This is the variable we are trying to predict

R-squared - variability of the data, explained by the regression model. Range: [0;1]

Adj. R-squared - variability of the data, explained by the regression model, considering the number of independent variables. Range: < 0; *could be negative, but a negative number is interpreted as 0*

F-statistic - evaluates the overall significance of the model (if at least 1 predictor is significant, F-statistic is also significant). The lower the F-statistics, the closer to a non-significant model

Prob (F-statistic) - P-value for F-statistic

- ***A coefficients table***

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.2750	0.409	0.673	0.503	-0.538	1.088
SAT	0.0017	0.000	7.487	0.000	0.001	0.002

std err = standard error – shows the accuracy of the prediction for each variable

→ Coefficient of the intercept, b_0; sometimes we refer to this variable as *constant* or *bias* (as it 'corrects' the regression equation with a constant value)

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.2750	0.409	0.673	0.503	-0.538	1.088
SAT	0.0017	0.000	7.487	0.000	0.001	0.002

Coefficient of the independent variable i: b_i; this is usually the most important metric – it shows us the relative/absolute contribution of each independent variable of our model

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.2750	0.409	0.673	0.503	-0.538	1.088
SAT	0.0017	0.000	7.487	0.000	0.001	0.002

P-value of t-statistic; The t-statistic of a coefficient shows if the corresponding independent variable is significant or not

- ***Additional tests***

Omnibus:	12.839	**Durbin-Watson:**	0.950
Prob(Omnibus):	0.002	**Jarque-Bera (JB):**	16.155
Skew:	-0.722	**Prob(JB):**	0.000310
Kurtosis:	4.590	**Cond. No.**	3.29e+04

A way for detecting autocorrelation (a violation of the fourth OLS assumption)

Decomposition of variability

Variance - a measure of *variability* calculated by taking the average of squared deviations from the mean. It describes how far the observed values differ from the average of predicted values

- Explained variability - variability explained by the explanatory variables used in our regression
- Unexplained variability - variability explained by other factors that are not included in the model

- Total variability - variability of the outcome variable

The total variability of the data set is equal to the variability explained by the regression line plus the unexplained variability, known as *error*.

- Sum of squares total (SST) = Total sum of squares (TSS) – measures the total variability of the dataset

  SST = $\sum_{i=1}^{n}(y_i - \bar{y})^2$

  y_i – observed dependent variable

  $\bar{y}$ - mean of the dependent variable

- Sum of squares regression (SSR) = Explained sum of squares (ESS) – measures the explained variability by the regression line

  SSR = $\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$

  $\hat{y}_i$- the predicted value of the dependent variable

  $\bar{y}$ - mean of the dependent variable

- Sum of squares error (SSE) = Residual sum of squares (RSS) – measures the unexplained variability by the regression

  SSE = $\sum_{i=1}^{n} {e_i}^2$

  e_i – the difference between the actual value of the dependent variable and the predicted value

  $e_i = y_i - \hat{y}_i$

  Total Variability = Explained variability + Unexplained variability:

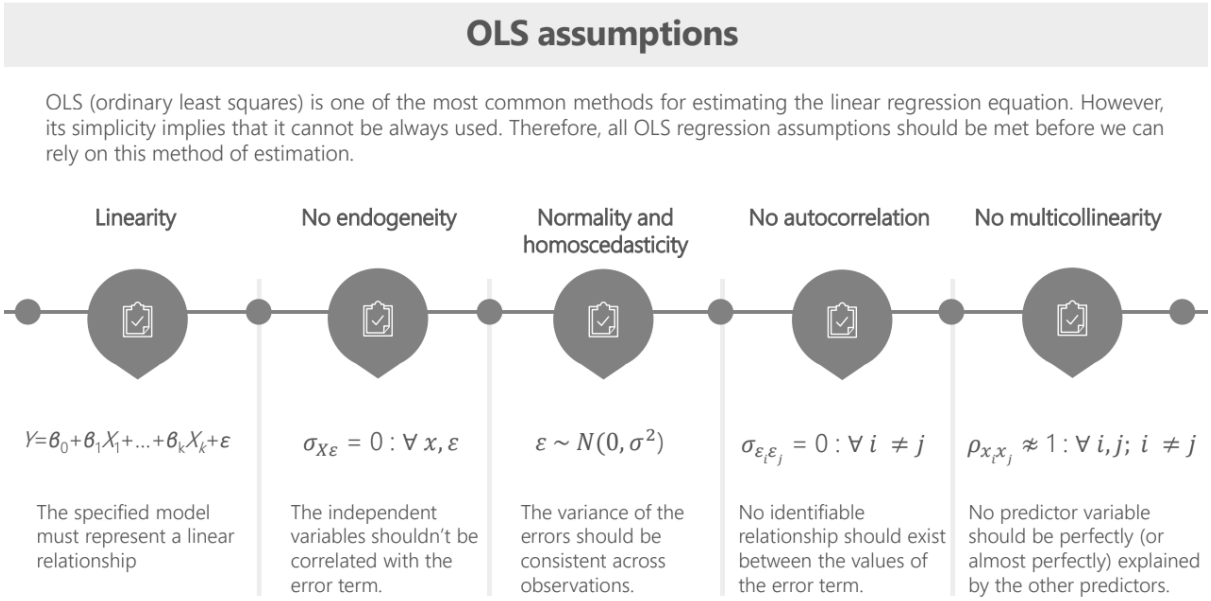
SST = SSR + SSE

↑SSR ↓SSE

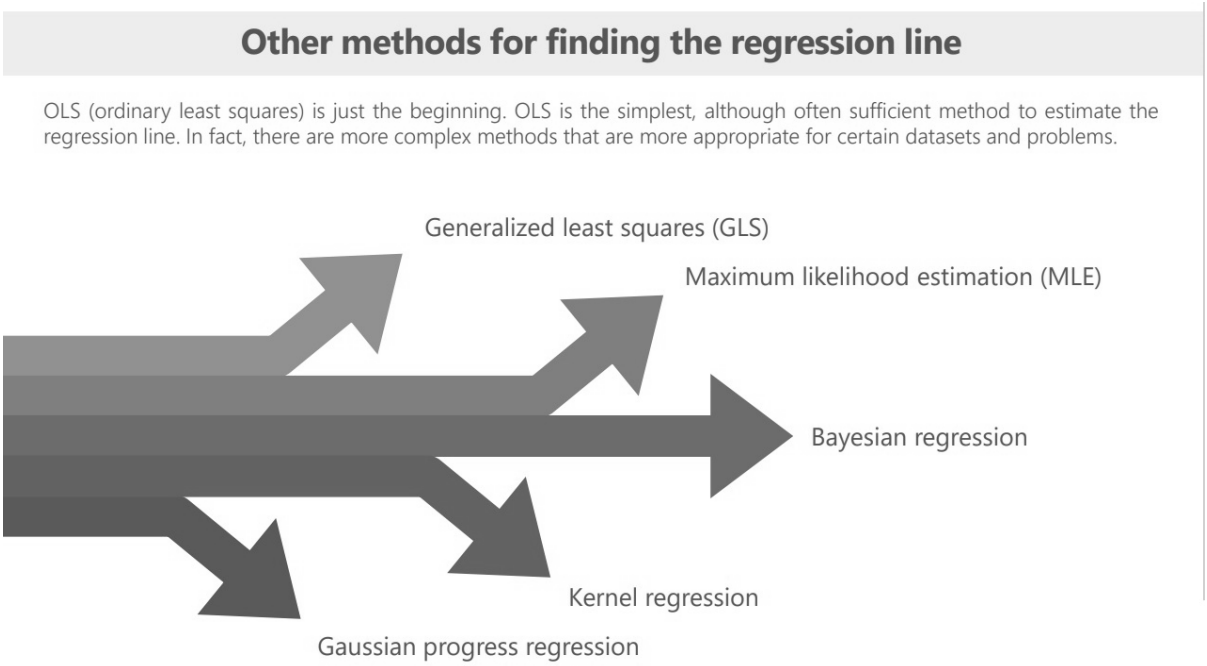
1.6 OLS

OLS or the ordinary least squares is the most common method to do estimate of the linear regression equation. “Least squares” stands for the minimum squares error, or SSE. This method aims to find the line, which minimizes the sum of the squared errors.

$$S(b) = \sum_{i=1}^{n} (y - xb)^2$$

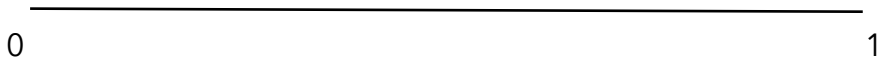


There are other methods for determining the regression line. They are preferred in different contexts.



1.7 R-squared

R-squared - a measure describing how powerful a regression is.



R-squared visual interpretation

0 - The regression explains none of the variability

1- The regression explains all of the variability

R-squared is a relative measure and takes values ranging from 0 to 1. An R squared of zero means your regression line explains none of the variability of the data. An R squared of 1 would mean your model explains the entire variability of the data. Unfortunately, regressions explaining the entire variability are rare. What you will usually observe is values ranging from 0.2 to 0.9.

1.8 Multiple linear regression

Population model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon$$

Multiple regression equation:

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots + + b_k x_k$$

$\hat{y}$ – inferred value

b_0 – intercept

$x_1 \ldots x_k$ – independent variables

$b_1 \ldots b_k$ – corresponding coefficients

- Addresses the higher complicity of a problem
- Contains many more independent variables
- It is not about the best fitting line as there is no way to represent the result graphically. Instead, it is about the best fitting *model*

  ➔ min SSE

SST = SSR + SSE

SSR SSE

1.9 Adjusted R-squared

Adjusted R-squared - measures how much of the total variability is explained by our model, considering the number of variables. The adjusted R squared is always smaller than the R squared, as it penalizes excessive use of variables. It is the basis for comparing models.

Multiple linear regression and adjuster R-squared in Python

Steps:

- Importing the relevant libraries
- Loading the data
- Defining the dependent and the independent variables following the regression equation -
- Regression itself
    - Adding a constant
    - Fitting the model according to the OLS method with a dependent variable *y* and an independent variable *x, which can contain multiple values – for instance a DataFrame of Series*
- Printing a summary of the regression

1.10 F-test

The F-statistic is used for testing the overall significance of the model.

F-test:

H_0: $\beta_1 = \beta_2 = \ldots = \beta_k = 0$

H_1: $at\ least\ one\ \beta_i \neq 0$

If all betas are 0, then none of the Xs matter

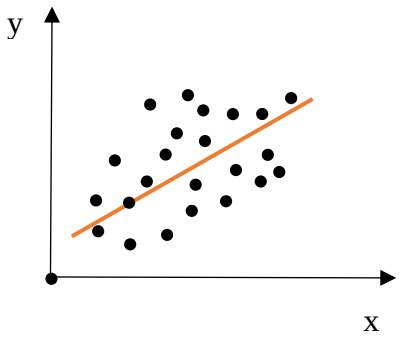
➔ our model has no merit

The lower the F-statistic, the closer to a non-significant model.

1.11 OLS Assumptions: Linearity

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon$$

The linear regression is the simplest non-trivial relationship.



A linear relationship between two variables

The easiest way is to verify if the relationship between two variables is linear is to choose an independent variable X one and plot it against the depended Y on a scatter plot. If the data points form a pattern that looks like a straight line, then a linear regression model is suitable.

If the relationship is non-linear, you should not use the data before transforming it appropriately.

Fixes:

- run a non-linear regression
- exponential transformation
- log transformation

1.12 OLS Assumptions: No endogeneity

No endogeneity refers to the prohibition of a link between the independent variables and the errors, mathematically expressed in the following way:

$$\sigma_{X\varepsilon} = 0 : \forall\ x, \varepsilon$$

The error (the difference between the observed values and the predicted values) is correlated with the independent values. This is a problem referred to as *omitted variable bias*. Omitted variable bias is introduced to the model when you forget to include a relevant variable.

As each independent variable explains *y*, they move together and are somewhat correlated. Similarly, *y* is also explained by the *omitted variable*, so they are also correlated. Chances are that the omitted variable is also correlated with at least one independent *x*. but it is not included it as a regressor. Everything that you don't explain with your model goes into the error. So, the error becomes correlated with everything else.

1.13 OLS Assumptions: Normality and homoscedasticity

$$\varepsilon \sim N(0, \sigma^2)$$

- Normality = *N* – assuming the error is normally distributed
- Zero mean - if the mean is not expected to be zero, then the line is not the best fitting one. However, having an intercept solves that problem, so in real-life it is unusual to violate this part of the assumption.
- Homoscedasticity = equal variance - the error terms should have equal variance one with the other

- Prevention:
    - Looking for omitted variable bias
    - Looking for outliers
    - Log transformation – creating a semi-log or a log-log model

1.14 OLS Assumptions: No autocorrelation

No autocorrelation = no serial correlation

$$\sigma_{\varepsilon_i \varepsilon_j} = 0 : \forall\, i \neq j$$

Errors are assumed to be uncorrelated.

Serial correlation is highly unlikely to be found it in data taken at one moment of time, known as *cross-sectional data*. However, it is very common in *time series data*.

Prevention:

- Dublin-Watson test:
    - 2 – no autocorrelation
    - < 1 and >3 – cause an alarm

When in the presence of autocorrelation - avoid the linear regression model!

Alternatives:

- Autoregressive model
- Moving average model
- Autoregressive moving average model

- Autoregressive integrated moving average model

1.15 OLS Assumptions: No multicollinearity

$$\rho_{x_i x_j} \approx 1 : \forall\, i, j;\ i \neq j$$

We observe multicollinearity when two or more variables have a high correlation. This poses a problem to our model.

Multicollinearity is a big problem but is also the easiest to notice. Before creating the regression, find the correlation between each two pairs of independent variables, and you will know if a multicollinearity problem may arise.

***Fixes*:**

- Dropping one of the two variables
- Transforming them into one (e.g. average price)
- Keeping them both while treating them with extreme caution

The correct approach depends on the research at hand.

1.16 Dealing with categorical data – Dummy variables

Dummy – an imitation of *categories* with *numbers*

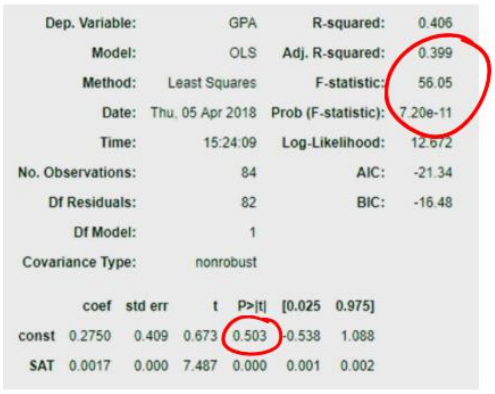
Example:

Attendance	
Categorical data	Numerical data
Yes	1

No 0

In regression analysis, a dummy is a variable that is used to include categorical data into a regression model.

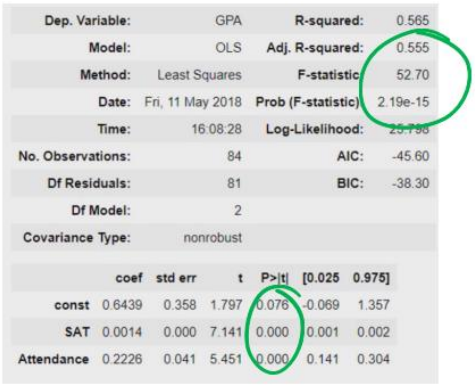
Without categorical data:



Dep. Variable:	GPA	R-squared:	0.406
Model:	OLS	Adj. R-squared:	0.399
Method:	Least Squares	F-statistic:	56.05
Date:	Thu, 05 Apr 2018	Prob (F-statistic):	7.20e-11
Time:	15:24:09	Log-Likelihood:	12.672
No. Observations:	84	AIC:	-21.34
Df Residuals:	82	BIC:	-16.48
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.2750	0.409	0.673	0.503	-0.538	1.088
SAT	0.0017	0.000	7.487	0.000	0.001	0.002

With categorical data:



Dep. Variable:	GPA	R-squared:	0.565
Model:	OLS	Adj. R-squared:	0.555
Method:	Least Squares	F-statistic:	52.70
Date:	Fri, 11 May 2018	Prob (F-statistic)	2.19e-15
Time:	16:08:28	Log-Likelihood:	25.798
No. Observations:	84	AIC:	-45.60
Df Residuals:	81	BIC:	-38.30
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.6439	0.358	1.797	0.076	-0.069	1.357
SAT	0.0014	0.000	7.141	0.000	0.001	0.002
Attendance	0.2226	0.041	5.451	0.000	0.141	0.304

1.17 Underfitting and overfitting

Underfitting model	Overfitting model	Good model
Doesn't capture the logic of the data	The training model has focused on the particular training set so much, that it captures all the noise and "misses the point"	Captures the underlying logic of the data
Low train accuracy	High train accuracy	High train accuracy
Low test accuracy	Low test accuracy	High test accuracy

Comparison between an overfitted model, an underfitted model, and a good model

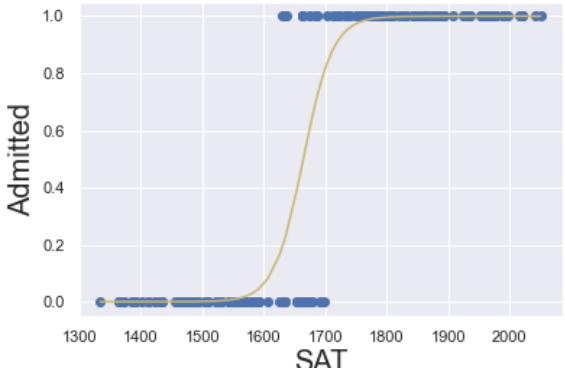
1.18 Training and testing

We split the data into *training* and *testing* parts and we *train* the model on the *training* dataset but *test* it on the *testing* dataset. The goal is to avoid the scenario where the model learns to predict the training data very well but fails when given new samples

- Generating data we are going to split
- Splitting the data
- Exploring the results

Section 2: Logistic Regression

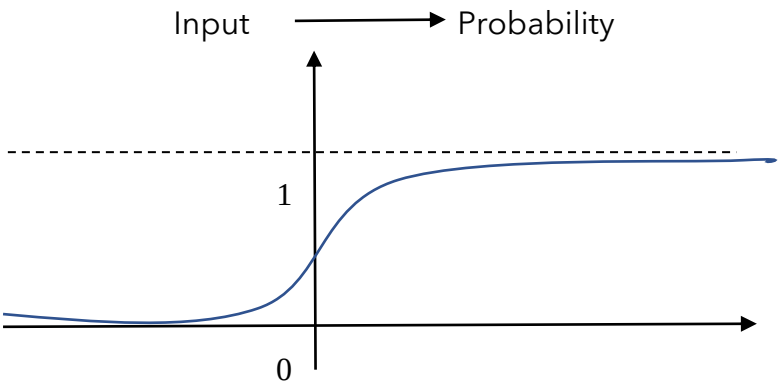
A logistic regression implies that the possible outcomes are not *numerical*, but rather – *categorical*. In the same way that we include categorical predictors into a linear regression through dummies, we can predict categorical outcomes through a logistic regression.



Logistic regression

2.1 Logistic vs logit function

The main difference between logistic and linear regressions is the *linearity assumption* – logistic regressions are *non-linear* by definition. The logistic regression predicts the probability of an event occurring. Thus, we are asking the question: given input data, what is the probability of a student being admitted?



Logistic function

Logistic regression model:

$$p(X) = \frac{e^{(\beta_0+\beta_1 x_1+\ldots+\beta_k x_k)}}{1 + e^{(\beta_0+\beta_1 x_1+\ldots+\beta_k x_k)}}$$

Exponential of a linear combination of inputs and coefficients, divided by, one, plus, the same exponential.

After transforming, the expression looks like this:

LogIt regression model:

$$Odds \rightarrow \frac{p(X)}{1 - p(X)} = e^{(\beta_0+\beta_1 x_1+\ldots+\beta_k x_k)}$$

The probability of the event occurring, divided by the probability of the event **NOT** occurring equals the exponential from above.

$$log(odds) = \beta_0 + \beta_1 x_1+\ldots+\beta_k x_k$$

Linear regression is the basic of logistic regression.

2.2 Building a logistic regression

Steps when working with statsmodels.api:

- Adding a constant
- Applying the *sm.Logit()* method - takes as arguments the dependent variable and the independent variable
- Fitting the regression

2.3 Understanding the tables

Logit Regression Results

Dep. Variable:	Admitted	**No. Observations:**	168
Model:	Logit	**Df Residuals:**	166
Method:	MLE	**Df Model:**	1
Date:	Fri, 11 Feb 2022	**Pseudo R-squ.:**	0.7992
Time:	15:55:37	**Log-Likelihood:**	-23.145
converged:	True	**LL-Null:**	-115.26
Covariance Type:	nonrobust	**LLR p-value:**	5.805e-42

Method MLE = Maximum likelihood estimation

- Likelihood function – a function which estimates how likely it is that the model at hand describes the real underlying relationship of the variables. The bigger the likelihood function, the higher the probability that our model is correct.

MLE tries to maximize the likelihood function

LL-Null = Log likelihood-null – the log-likelihood of a model which has no independent variables

LLR p-value = Log likelihood ratio – measures if our model is statistically different from LL-null, a.k.a. a useless model

Pseudo R-squ. = Pseudo R-squared – this measure is mostly useful for comparing variations of the same model. Different models will have completely different and incomparable Pseudo R-squares.

- AIC
- BIC
- McFadden's R-squared

	coef	std err	z	P>|z|	[0.025	0.975]
const	-69.9128	15.737	-4.443	0.000	-100.756	-39.070
SAT	0.0420	0.009	4.454	0.000	0.024	0.060

$$\triangle odds = e^{b_k}$$

For a unit change in a variable, the change in the odds equals the exponential of the coefficient. That exactly is what provides us with a way to interpret the coefficients of a logistic regression.

Binary predictors in a logistic regression - in the same way we create dummies for a linear regression, we can use binary predictors in a logistic regression.

2.4 Calculating the accuracy of the model

Accuracy

```
np.set_printoptions(formatter={'float': lambda x: "{0:0.2f}".format(x)})
results_log.predict()
```

```
array([0.00, 1.00, 1.00, 0.23, 0.02, 0.99, 1.00, 1.00, 1.00, 0.01, 1.00,
       1.00, 0.76, 0.00, 0.60, 1.00, 0.11, 0.12, 0.51, 1.00, 1.00, 1.00,
       0.00, 0.01, 0.97, 1.00, 0.48, 0.99, 1.00, 0.99, 0.00, 0.83, 0.25,
       1.00, 1.00, 1.00, 0.31, 1.00, 0.23, 0.00, 0.02, 0.45, 1.00, 0.00,
       0.99, 0.00, 0.99, 0.00, 0.00, 0.01, 0.00, 1.00, 0.92, 0.02, 1.00,
       0.00, 0.37, 0.98, 0.12, 1.00, 0.00, 0.78, 1.00, 1.00, 0.98, 0.00,
       0.00, 0.00, 1.00, 0.00, 0.78, 0.12, 0.00, 0.99, 1.00, 1.00, 0.00,
       0.30, 1.00, 1.00, 0.00, 1.00, 1.00, 0.85, 1.00, 1.00, 0.00, 1.00,
       1.00, 0.89, 0.83, 0.00, 0.98, 0.97, 0.00, 1.00, 1.00, 0.03, 0.99,
       0.96, 1.00, 0.00, 1.00, 0.01, 0.01, 1.00, 1.00, 1.00, 0.00, 0.00,
       0.02, 0.33, 0.00, 1.00, 0.09, 0.00, 0.97, 0.00, 0.75, 1.00, 1.00,
       0.01, 0.01, 0.00, 1.00, 0.00, 0.99, 0.57, 0.54, 0.87, 0.83, 0.00,
       1.00, 0.00, 0.00, 0.00, 1.00, 0.04, 0.00, 0.01, 1.00, 0.99, 0.52,
       1.00, 1.00, 0.05, 0.00, 0.00, 0.00, 0.68, 1.00, 1.00, 1.00, 1.00,
       1.00, 0.00, 1.00, 1.00, 0.04, 1.00, 0.02, 1.00, 0.99, 0.97, 0.94,
       0.01, 0.00, 0.00])
```

```
np.array(data['Admitted'])
```

```
array([0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1,
       0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0,
       0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0], dtype=int64)
```

```
results_log.pred_table()
```

```
array([[69.00, 5.00],
       [4.00, 90.00]])
```

Confusion matrix

For 69 observations, the model predicted 0 when the true value was 0.

For 90 observations, the model predicted 1, and it actually was 1.

These cells indicate in how many cases the model did its job well.

In 4 cases, the model predicted 0, while it was 1.

In 5 cases, the regression predicted 1 while it was 0.

*The most important metric we can calculate from this matrix is the **accuracy** of the model.*

```
cm_df = pd.DataFrame(results_log.pred_table())
cm_df.columns = ['Predicted 0','Predicted 1']
cm_df = cm_df.rename(index={0:'Actual 0',1:'Actual 1'})
cm_df
```

	Predicted 0	Predicted 1
Actual 0	69.0	5.0
Actual 1	4.0	90.0

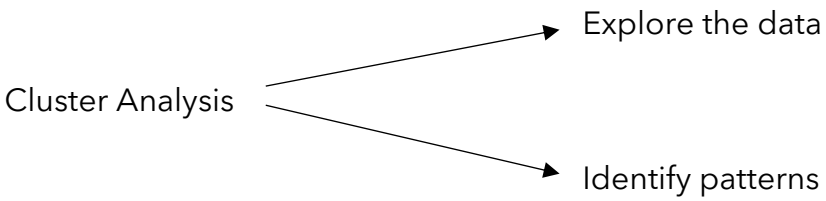
```
cm = np.array(cm_df)
accuracy_train = (cm[0,0]+cm[1,1])/cm.sum()
accuracy_train
```

```
0.9464285714285714
```

In 69 plus 90 of the cases, the model was correct. In 4 plus 5 of the cases, the model was incorrect. Overall, the model made an accurate prediction in 159 out of 168 cases. That gives us a 159 divided by 168, which is 94.6% accuracy.

Section 3: Cluster Analysis

Cluster analysis is a multivariate statistical technique that groups observations on the basis some of their features or variables that they are described by. The goal of clustering is to maximize the similarity of observations *within* a cluster and maximize the dissimilarity *between* clusters.

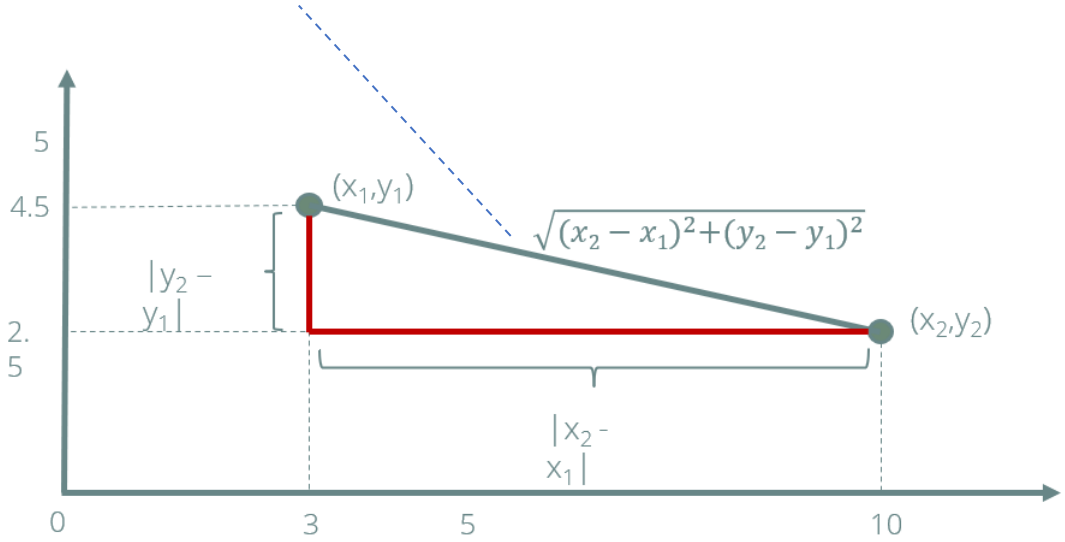


Classification	Clustering
Model (Inputs) > Outputs > Correct values	Model (Inputs) > Outputs > ?
Predicting an output category, given input data	Grouping data points together based on similarities among them

Comparison between classification and clustering

- Measuring the distance between two data points
- Defining the term 'centroid'

Euclidean distance



Geometrical representation of the relationship between two points

2D space: d(A,B) = d(B,A) = $\sqrt{(x_2 - x_1 1)^2 + (y_2 - y_1)^2}$

3D space: d(A,B) = d(B,A) = $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)2}$

If the coordinates of A are (a_1, b_1,…, a_n) and of B are (b_1, b_2,… b_n):

N-dim space:

d(A,B) = d(B,A) = $\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2}$

Centroid – the mean position of a group of points

Section 4: K-Means Clustering

K-means is the most popular method for identifying clusters.

- Choosing the number of clusters

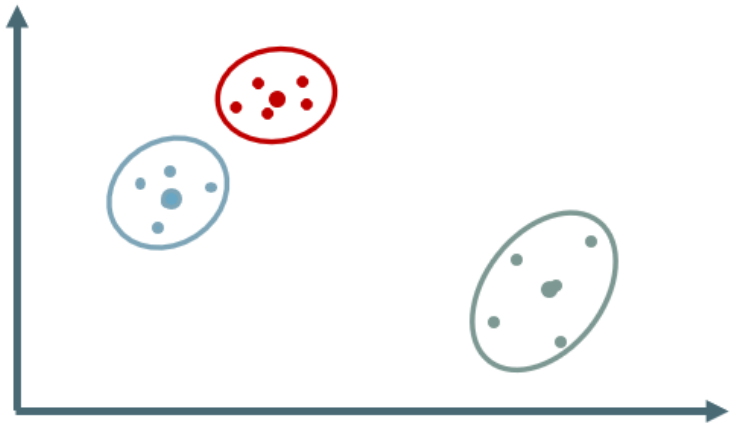
K – the number of clusters we are trying to identify

- Specifying the cluster seeds based on a prior knowledge on the given data

Seed – a starting centroid

- Assigning each data point to a centroid based on proximity (measured with Euclidean distance)
- Adjusting the centroids

Reassigning the data points and recalculate the centroids



Clusters

The goal is to minimize the distance between the points in a cluster (WCSS) and maximize the distance between the clusters.

WCSS - within-cluster sum of squares

4.1 Selecting the number of clusters

The Elbow method – a criterion for setting the proper number of clusters. It is about making the WCSS as low as possible, while still having a small number of clusters to reason about.

4.2 K-means clustering - pros and cons

Pros	Cons
Simple to implement	We need to pick K
Computationally efficient	Sensitive to initialization
Widely used	Sensitive to outliers
Always yields a result	Produces spherical solutions

Pros and cons of the K-means clustering

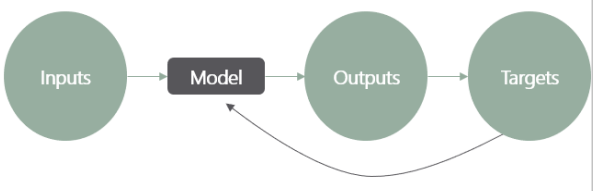
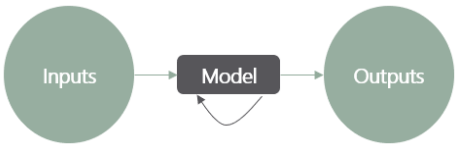
4.3 Standardization

The aim of standardization is to reduce the weight of higher numbers and increase that of lower ones. If we don't standardize, the range of the values serves as weights for each variable, and we are not taking advantage of the size data whatsoever. Therefore, it is a good practice to standardize the data before clustering.

When we should *not* standardize - as standardization is trying to put all variables on equal footing, in some cases, we don't need to do that. If we know that

one variable is inherently more important than another, then standardization shouldn't be used.

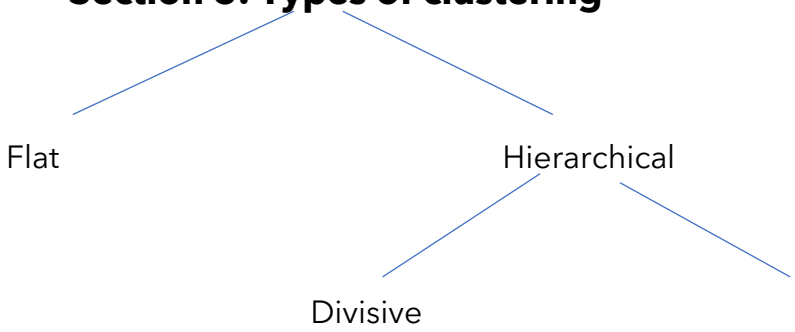
4.4 Relationship between clustering and regression

Classification	**Clustering**
Classification is a typical example of supervised learning	Cluster analysis is a typical example of unsupervised learning
It is used whenever we have input data and the desired correct outcomes (targets). We train our data to find the patterns in the inputs that lead to the targets	It is used whenever we have input data but have no clue what the correct outcomes are
With classification we need to know the correct class of each of the observations in our data, in order to apply the algorithm	Clustering is about grouping data points together based on similarities among them and difference from others
A typical example of classification is the logistic regression	A typical example of clustering is the K-means clustering
	

Comparisson between classification and clustering

In classification, we use the targets (correct values) to adjust the model to get better outputs. In clustering, there is no feedback loop, therefore, the model simply finds the outputs it deems best.

Section 5: Types of clustering



Types of clustering

Flat - with flat methods there is no hierarchy, but rather the number of clusters are chosen prior to clustering. Flat methods have been developed because hierarchical clustering is much slower and computationally expensive. Nowadays, flat methods are preferred because of the volume of data we typically try to cluster.

Hierarchical - historically, hierarchical clustering was developed first. An example clustering with hierarchy is the taxonomy of the animal kingdom. It is superior to flat clustering in the fact that it explores (contains) all solutions.

- **Divisive (top-down) -** with divisive clustering, we start from a situation where all observations are in the same cluster. Then, we split this big cluster into 2 smaller ones. Next, we continue with 3, 4, 5, and so on, until each observation is its separate cluster. To find the best split, we must explore all possibilities at each step.
- **Agglomerative (bottom-up) - w**hen it comes to agglomerative clustering, the approach is bottom up. To find the combination of observations into a cluster, we must explore all possibilities at each step.

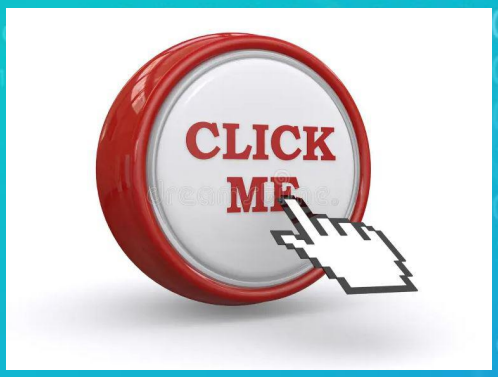
5.1 Dendrogram

Pros:

- Hierarchical clustering shows all the possible linkages between clusters
- It's easy to understand the data
- No need to define the number of clusters (like with K-means)
- There are many methods performing hierarchical clustering

Cons:

- The biggest con is *scalability*
- It is computationally expensive – the more observations, the slower it gets



Other Part of this Series

1. Machine Learning Statistics
2. Inroduction to Python
3. Machine Learning with Decision Tree and random Forest
4. Decision Tree
5. Machine Learning in Python
6. Python in Finance
7 Data Science

Click Here to Download

Iliya Valchanov
Ivan Manov

Email: team@365datascience.com

365 DataScience