Logistic regression analysis

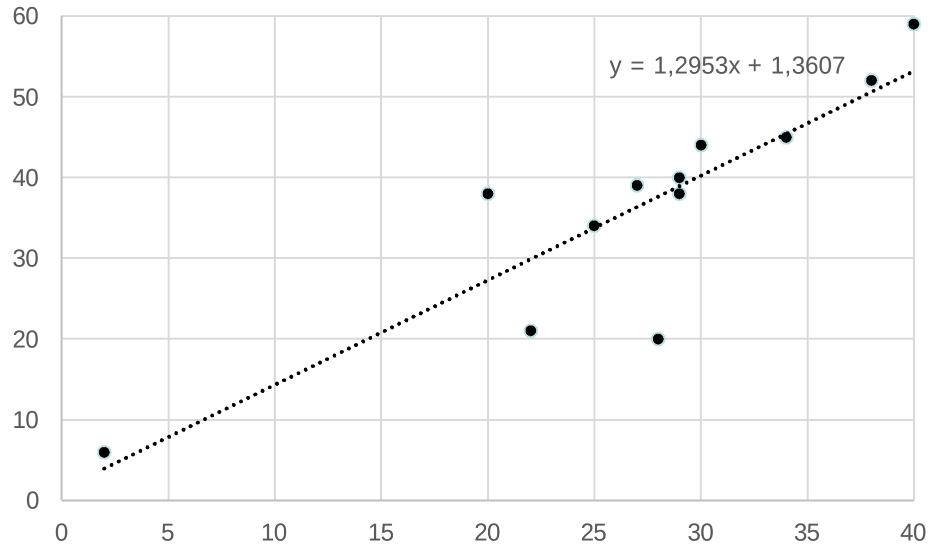
Predicting occurrences of events

Learning goals

1. Learn to apply a logistic regression model to predict an outcome of an event.
2. Learn to carry out the analysis in Python environment.

Linear vs. logistic regression

- Earlier we focused on linear regression.
  - In linear regression, the response variable is a continuous variable that can ideally vary in the interval $]-\infty, \infty[$.
  - In practice, the applicability of the model is limited.



Linear vs. logistic regression

- In logistic regression, the response variable is not a numerical variable but a binary class variable (yes/no).
- The goal of logistic regression analysis is to predict whether an event occurs or no.
- Technically the target of prediction is the probability p of an event.
  - Example: predict the probability for a subject experiencing a stroke, or, whether he/she will buy a car.

Logistic regression as a classifier

- As a consequence of estimating the probability of an event, logistic regression model can be used as a binary classifier.
  - Rule: If the probability of an event is estimated to be greater than 0.5, classify as ”yes”; otherwise ”no”.
  - Such binary classification based on the outcomes loses information on the uncertainty.

Probability as a response variable

- Recall that the equation in a generalized linear model is of form:

$$y_i = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

- The probability p of an event would be a bad choice for a response variable y, since its value is always in the interval [0,1].
- A linear model is designed to provide predictions within an unlimited range.
- Key question: how could the response variable be transformed in such a way that the range becomes [0,1]?

Odds

- Odds describes the ratio between an event and its complement.
  - Denote the probability of an event by p.
  - Odds is then defined as $\frac{p}{1-p}$.
- Let's assume that the probability of a person winning a running contest is 0,15.
- The odds are $\frac{0{,}15}{1-0{,}15} = \frac{0{,}15}{0{,}85} \approx 0{,}176$.
  - The probability of a win is 0,176 times as big as that of a loss.
  - Or, expressed in reverse terms, the probability of a loss is appxroximately
  - 5,67-fold in comparison to that of a win.
- The range of odds is $[0, \infty[$.
  - It is still constrained from the lower edge.
  - In addition, the values of interest are often ”packed” close to the zero.

Logit

- If we take a natural logarithm of the odds, we have made a logit transformation for the probability p.
- Thus, logit transformation is expressed as:

  $\ln \frac{p}{1-p}$

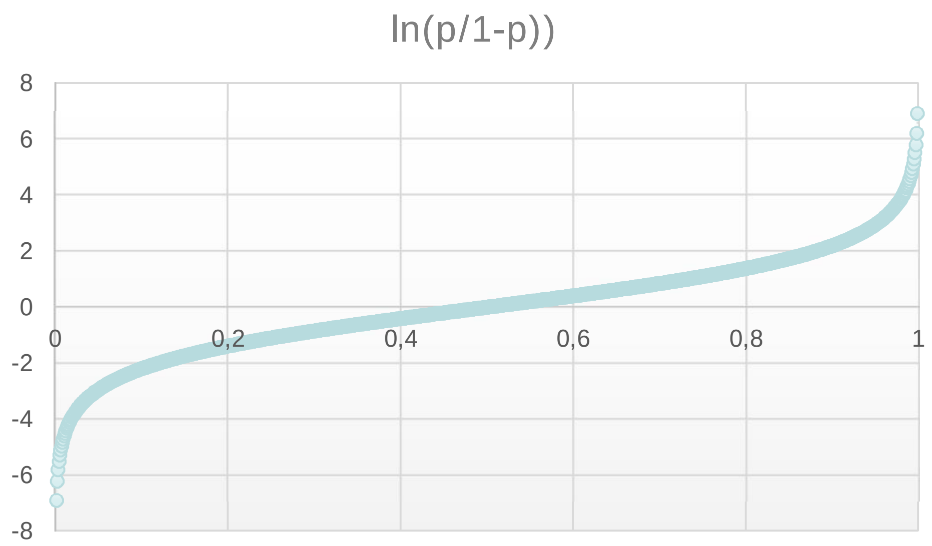
- E.g. the probability p = 0,15 of a win corresponds to the logit value:
- $\ln \frac{0{,}15}{1-0{,}15} \approx -1{,}735$

Logit

p	1-p	ln(p/1-p))
0,001	0,999	-6,90675
0,002	0,998	-6,21261
0,003	0,997	-5,80614
0,004	0,996	-5,51745
0,005	0,995	-5,2933
0,006	0,994	-5,10998
0,007	0,993	-4,95482
0,008	0,992	-4,82028
0,009	0,991	-4,70149
0,01	0,99	-4,59512
0,011	0,989	-4,4988
0,012	0,988	-4,41078
0,013	0,987	-4,32972

0,498	0,502	-0,008
0,499	0,501	-0,004
0,5	0,5	0
0,501	0,499	0,004
0,502	0,498	0,008

0,998	0,002	6,212606
0,999	0,001	6,906755



- Note that the value of the logit function is >0, when p>0,5.

Logistic regression model

- A logistic regression model is of form:

$$\ln \frac{p}{1-p} = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

- The transformed response variable can vary between $]-\infty, \infty[$.
  - The corresponding probabilities p are in range $[0,1]$.
- If the predicted value of a logit response variable is positive:
  - The prediction for $p > 0{,}5$
  - The event is predicted to happen (prediction ”1”)
- Likewise, a negative predicted value of the logit response variable corresponds to predicted non-occurrence of an event (predicted ”0”).

Limitations

- Many of the limitations of linear regression can be relaxed for logistic regression. For example:
  - The relationships needs not be linear.
  - The residuals don’t need to be normally distributed.
  - The variances of the residuals need not be constant.
- However:
  - The response variable needs to be binary.
  - The observations should be independent of each other.
  - There should not be much collinearity (dependence between variables).
  - The log odds of the response variable should be linearly related to the explanatory variables.

Source:
http://www.statisticssolutions.com/assumptions-of-logistic-regression/

Limitations

- There is no simple way to calculate the relative importances of the variables.
  - Regression coefficients from normalized data don’t provide the answer.
  - This is due to the inherent non-linearities in the model

Log loss function

Actual	Predicted P(X=1)	Predicted P(X=actual)	Log of P(X=actual)
0	0,9	0,1	-1
1	0,62	0,62	-0,207608311
1	0,93	0,93	-0,031517051
1	0,42	0,42	-0,37675071
0	0,13	0,87	-0,060480747
0	0,21	0,79	-0,102372909
1	0,78	0,78	-0,107905397
0	0,14	0,86	-0,065501549

Logloss
0,244017

- If a set of regression coefficients is fixed, the value of logit-transformed response variable can be computed for each observation.
- This, in turn, can be transformed to a probability of belonging to class 1. That is, $P(X=1)$ by inverse logit transformation.
- Above, $P(X = actual)$, is $1 - P(X=1)$ for actual class 0, and $P(X=1)$ for actual class 1.
- Log loss function is the negative mean of logarithms of $P(X = actual)$ values.

Python example

- The Stroke data contains information about people who have experienced a stroke.
- The patients have been monitored for a one-year period.
- During the monitoring period, some patients have suffered a second stroke.
  - The occurrence of the second stroke is the response variable.
  - Ideally, the model could be used as a predictive tool for spotting the individuals with a high risk for the second stroke.

Python example

- The Python source code can be found in the **Methods / Data / Stroke (demo)** folder in the course’s Oma workspace.

`sklearn` specifics

- Logistic regression can also be used as a multiclass classifier.
- By default **`sklearn`** uses one-vs-rest (OvR) scheme.
  - A binary classifier is built for each value of the categorical response variable.
  - For each binary classifier, all remaining values of the response variable are lumped together.
  - Finally, for each observation, the classifier that provides a classification with a highest confidence score, outputs the final class.
- Regularization is applied by default.
  - It adds a penalty for non-zero regression coefficients

One-vs-Rest classification

	Classifier-specific response variables			
Orig. resp. variable	C1_class	C2_class	C3_class	C4_class
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
2	0	1	0	0
2	0	1	0	0
1	1	0	0	0
4	0	0	0	1
3	0	0	1	0

- For logistic regression, **sklearn** implements the OvR schema automatically, under the hood.

Regularization

- Regularization forces the regression coefficients towards zero.
  - In effect, that shrinks the number of exploratory variables in the model.
  - It may make the model less prone to overfitting.
- Idea of regularization: add penalty term to the log loss function:

$$logloss' = logloss + \lambda \sum_{j=1}^{p} \beta_j^2$$

- This is called L2 regularization (or Ridge regression).