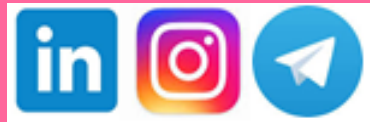


# Machine Learning 7

## Logistic Regression



behzad.amanpour





# Logistic Regression

- Predicts the probability of a categorical dependent variable.
- The dependent variable is a binary variable that contains data coded as 1 (yes) or 0 (no).
- Logistic regression model predicts  $P(Y=1)$  as a function of  $X$ .

$$\text{Ln}\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k$$

Linear Regression:  $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$

<https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8>



# Sklearn Logistic Regression

The screenshot shows the official documentation for `sklearn.linear_model.LogisticRegression` on the scikit-learn website. The page features a navigation bar with links to 'Install', 'User Guide', 'API', 'Examples', 'Community', and 'More'. A sidebar on the left contains navigation links for 'Prev', 'Up', and 'Next', along with version information for 'scikit-learn 1.1.3' and a 'cite us' notice. The main content area displays the class name `sklearn.linear_model.LogisticRegression` in a large font, followed by its constructor signature: `class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)`. Below the signature is a '[source]' link. The text describes it as a 'Logistic Regression (aka logit, MaxEnt) classifier' and explains the 'multi\_class' options: 'ovr' for one-vs-rest and 'multinomial' for multinomial logistic regression.

sklearn.linear\_model.LogisticRegression

```
class sklearn.linear_model.LogisticRegression(penalty='l2', *,
dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1,
class_weight=None, random_state=None, solver='lbfgs', max_iter=100,
multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)
```

Logistic Regression (aka logit, MaxEnt) classifier.

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the 'multi\_class' option is set to 'ovr', and uses the cross-entropy loss if the 'multi\_class' option is set to 'multinomial'. (Currently the 'multinomial' option is



# Sklearn Logistic Regression

The screenshot shows the documentation for `sklearn.linear_model.LogisticRegression` on the scikit-learn website. The page includes the scikit-learn logo, navigation buttons (Prev, Up, Next), version information (scikit-learn 1.1.3), and a citation notice. The main content area is divided into two columns. The left column contains the class name and a snippet of code: `sklearn.linear_model.LogisticRegression`. The right column contains detailed documentation for the `LogisticRegression` class, including a note about `sample_weight`, a new parameter `class_weight='balanced'` introduced in version 0.17, and the `random_state` parameter. The `solver` parameter is also documented, listing the available solvers: `'newton-cg'`, `'lbfgs'`, `'liblinear'`, `'sag'`, and `'saga'`. A list of aspects to consider when choosing a solver is provided at the bottom.

Note that these weights will be multiplied with `sample_weight` (passed through the fit method) if `sample_weight` is specified.

*New in version 0.17: `class_weight='balanced'`*

**`random_state` : int, RandomState instance, default=None**  
Used when `solver == 'sag', 'saga' or 'liblinear'` to shuffle the data. See [Glossary](#) for details.

**`solver` : {'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}, default='lbfgs'**  
Algorithm to use in the optimization problem. Default is 'lbfgs'. To choose a solver, you might want to consider the following aspects:

- For small datasets, 'liblinear' is a good choice, whereas 'sag' and 'saga' are faster for large ones;
- For multiclass problems, only 'newton-cg', 'sag', 'saga'





# Sklearn Logistic Regression

The screenshot shows the documentation page for `sklearn.linear_model.LogisticRegression` on the scikit-learn website. The page includes the scikit-learn logo, navigation buttons (Prev, Up, Next), and the current version (1.1.3). The main content area is partially visible, showing the class name and some introductory text. A sidebar on the right contains a 'See also' section with a link to the User Guide, and a 'New in version' section with information about the Stochastic Average Gradient descent solver and the SAGA solver. A 'Changed in version' section notes that the default solver changed from 'liblinear' to 'lbfgs' in version 0.22. The 'max\_iter' parameter is highlighted with its default value of 100.

sklearn.linear\_model.LogisticRegression

scikit-learn 1.1.3

Other versions

Please [cite us](#) if you use the software.

sklearn.linear\_model.LogisticRegression

Examples using

sklearn.linear\_model.LogisticRegression

**See also:** Refer to the User Guide for more information regarding [LogisticRegression](#) and more specifically the Table summarizing solver/penalty supports.

*New in version 0.17:* Stochastic Average Gradient descent solver.

*New in version 0.19:* SAGA solver.

*Changed in version 0.22:* The default solver changed from 'liblinear' to 'lbfgs' in 0.22.

**max\_iter : int, default=100**

Maximum number of iterations taken for the solvers to converge.