

# Logistic regression: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2019

## Syntax

- Defining the logistic function:

```
def logistic(x):  
    """  
    np.exp(x) raises x to the exponential power e^x. e ~= 2.71828  
    """  
  
    return np.exp(x) / (1 + np.exp(x))
```

- Instantiating a logistic regression model:

```
from sklearn.linear_model import LogisticRegression  
  
linear_model = LogisticRegression()
```

- Training a logistic regression model:

```
logistic_model.fit(admissions[["gpa"]], admissions["admit"])
```

- Returning predicted probabilities for a column:

```
pred_probs = logistic_model.predict_proba(admission[["gpa"]])
```

## Concepts

- In classification, our target column has a finite set of possible values, which represent different categories a row can belong to.
- In binary classification, there are only two options for values:
  - `0` for the False condition.
  - `1` for the True condition.
- Categorical values are used to represent different options or categories. Classification focuses on estimating the relationship between the independent variables and the dependent categorical variable.

- One technique of classification is called logistic regression. While a linear regression model outputs a real number as the label, a logistic regression model outputs a probability value.
- The logistic function is a version of the linear function that is adapted for classification. Mathematically, the logistic function is represented as the following:

where  $e^x$  is the exponential transformation to transform all values to be positive, and  $\frac{1}{1 + e^{-x}}$  is the normalization transformation to transform all values between  $0$  and  $1$ .

## Resources

- [Documentation for the LogisticRegression class](#)
- [Documentation for the predict\\_proba method](#)



Takeaways by Dataquest Labs, Inc. - All rights reserved © 2019