**Logistic Regression**

1. <https://medium.com/@anishsingh20/logistic-regression-in-python-423c8d32838b>
2. <https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python>
3. <https://towardsdatascience.com/logistic-regression-python-7c451928efee> - important one

# **Advantages:**

* it doesn’t require high *computational power*
* is easily *interpretable*
* is used widely by **data analysts** and **data scientists**.
* is very easy to ***implement***
* it doesn’t require *scaling* of *features*
* it provides a ***probability*** *score* for *observations*.

**Disadvantages:**

* while working with ***Logistic regression*** you are not able to handle a large number of *categorical features/variables*.
* it is **vulnerable** to overfitting
* it can’t solve the *non-linear* problem with the *logistic regression model* that is why it requires a *transformation* of *non-linear* features
* **Logistic regression** will not perform well with the *independent(****X****)* variables that are not **correlated** to the *target(****Y****)* variable.

# **#KeyFeatures**

* ***Logistic regression*** *predicts* whether something is *True(1)*or *False(0)*instead, *predicting* something that is *continuous* like the size.
* It has an ***S-shaped*** line.
* We can take our ***Linear Regression Model*** and convert it into ***Logistic Regression Model*** with the help of **Sigmoid Function.**
* ***Logistic Regression’s*** *ability* to provide *probabilities* and *classify* new samples using ***continuous*** and ***discrete*** measurements makes it a popular **machine learning** method.

# **Implementation in Python: Jupyter Notebook**

* Handling\_Missing\_Data(logistic\_regression)
* Slides from SimpliLearn