# Derivatives | Gradients | Gradient Descent

**What Derivatives tell us?**

Derivatives tells us if a function is increasing or decreasing (curve is going upward or downward). If a function is increasing then its derivative is positive and if the function is decreasing then the derivative is negative.

**What Gradient tells us?**

Gradient tells us the steepness of a function. How fast the function is increasing or decreasing.

**What is Gradient Descent?**

It’s an approach to find the optimal value efficiently. It takes bigger steps where the result is far from the optimal and takes smaller steps when it closes to the optimal value. In order to achieve that it uses "derivative".

A parameter called "learning rate" defines how big those starting steps should be.

**What is Stochastic Gradient Descent**

For functions where lot of parameters are involved gradient descent approach can take long time. In those cases, stochastic gradient descent optimizes the process by randomly selecting subset of the data at every step rather than the full dataset.

# Syllabus

**In first module** we define a linear classifier in logistic regression.

**In second module** we're going to figure out how to learn the parameters (the coefficients of the classifiers) from data, and find the best model with **“gradient ascent”**.

**Third module:** Overfitting and regularization

**Fourth module: Descission tree:** Useful for non-linear data.

**Fifth module:** Overfitting for descission tree.

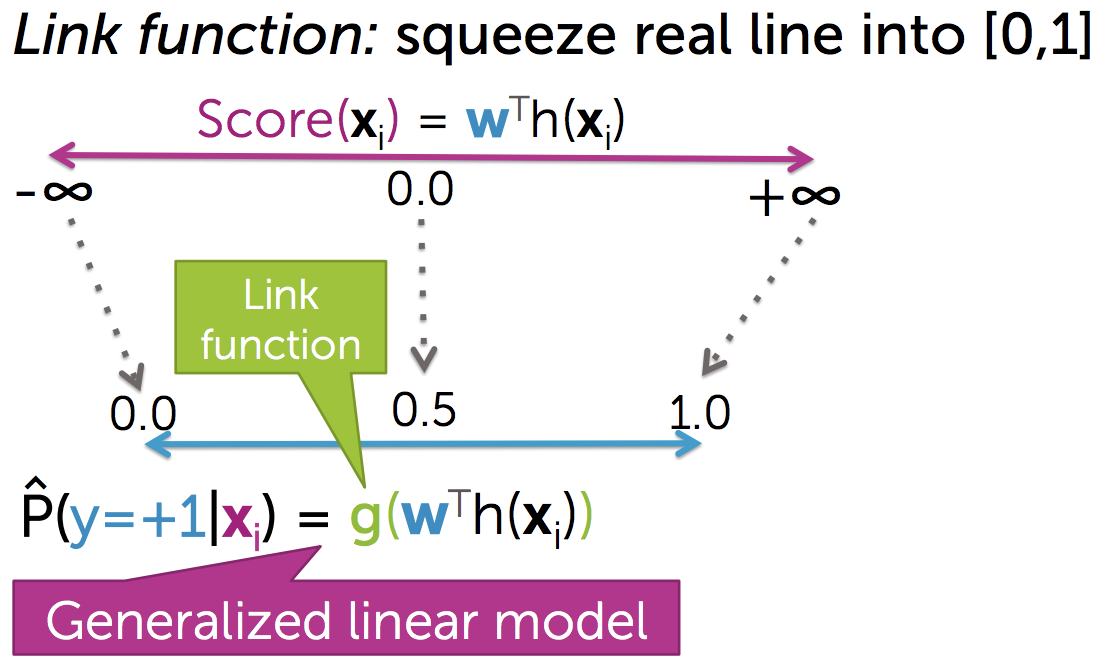
**Sixth module: Handling missing data:** How a descission tree can take into account missing data and accommodate?

**Seventh module: Boosting:** Combining a set of week learners to create a strong learner. **AdaBoost**.

Eightth module: Precission recall:

Nineth module: Scalability. Stocastic Gradient.

**Generalized Linear Model:** Squeeze real line into [0,1] using a link function.



# Logistic Regression

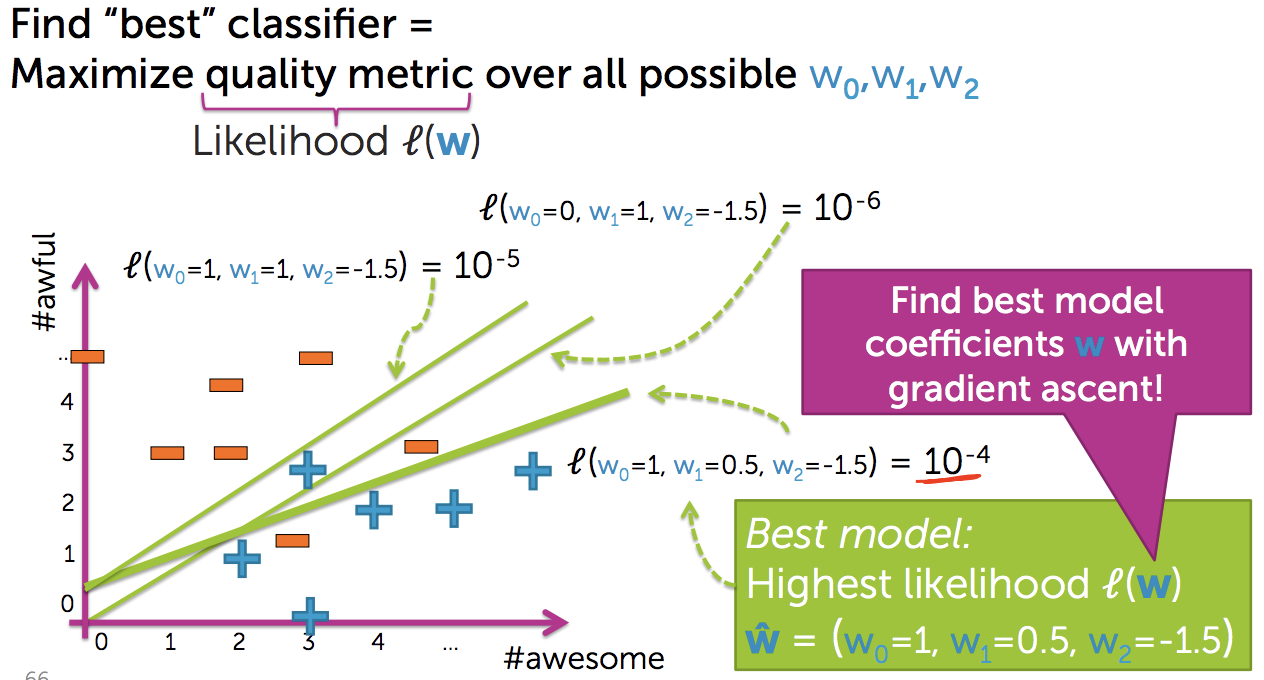
Logistic regression: It squeezes a score into [0,1] using a sigmoid function.

Logistic regression model: It takes a score as input that ranges from minus infinity to plus infinity (w transposed h of x i) and It pushes it through the sigmoid function to estimate the probability that y=+1 given xi and w.

**[Quality Matric]**

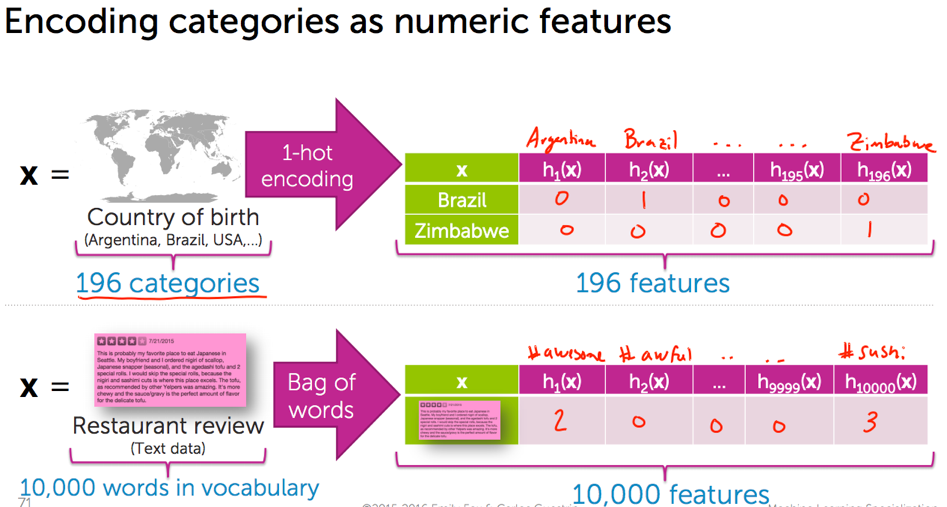
**Cost function**: Likelyhood: l(w)

Like Linear regression, we need to find the best line that classify two sets [+1, -1]. And we'll use a “**gradient ascent”** algorithm to find the set of parameters (w) that has the highest likelihood to produce the best quality.

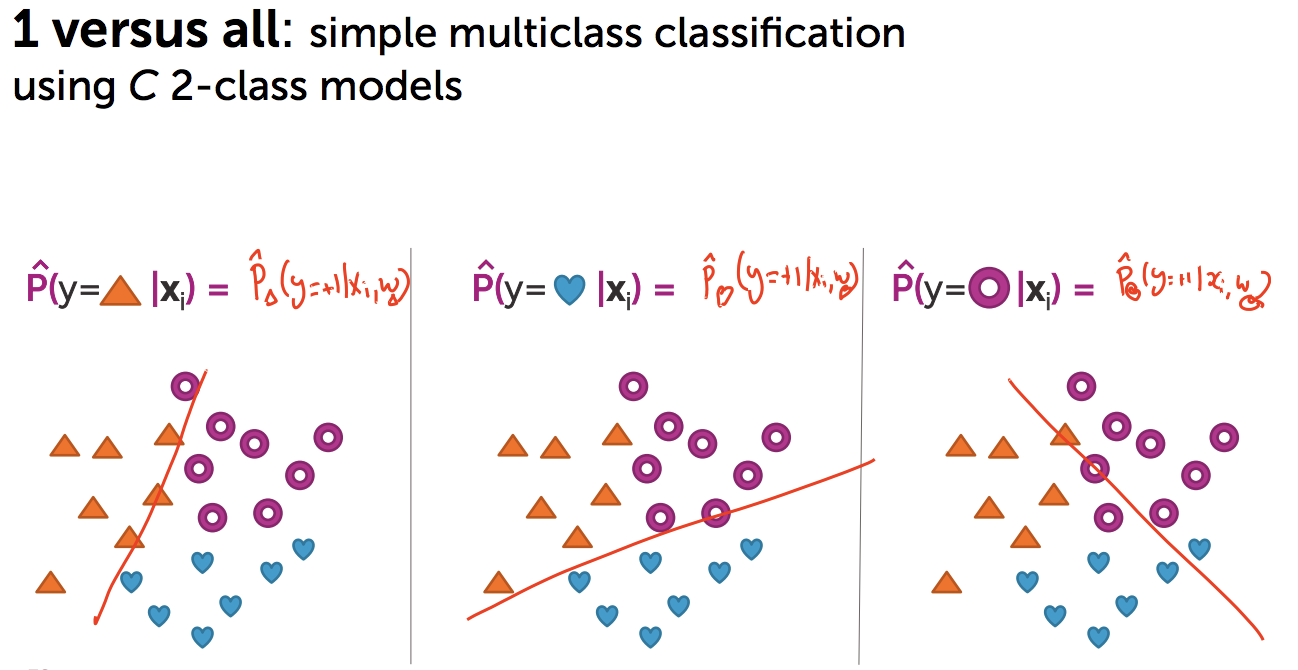


Non numeric features (categorical values): We can’t simply multiply coefficients with categorical values like numerical values. So how to handle categorical values? Answer is: “Encoding”. e.g. one-hot encoding, bag of words.

By encoding we transform categorical values into numerical values then multiply with coefficient.



**Multiclass classification**: an example of it is “one versas all”.

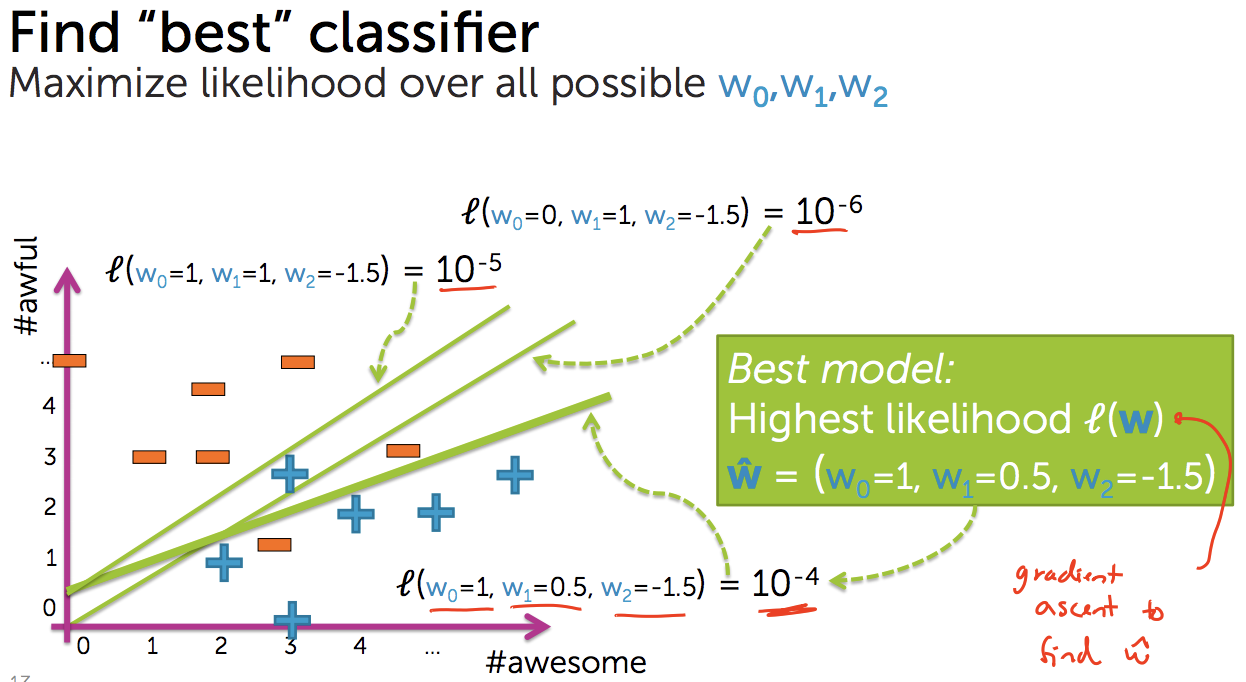


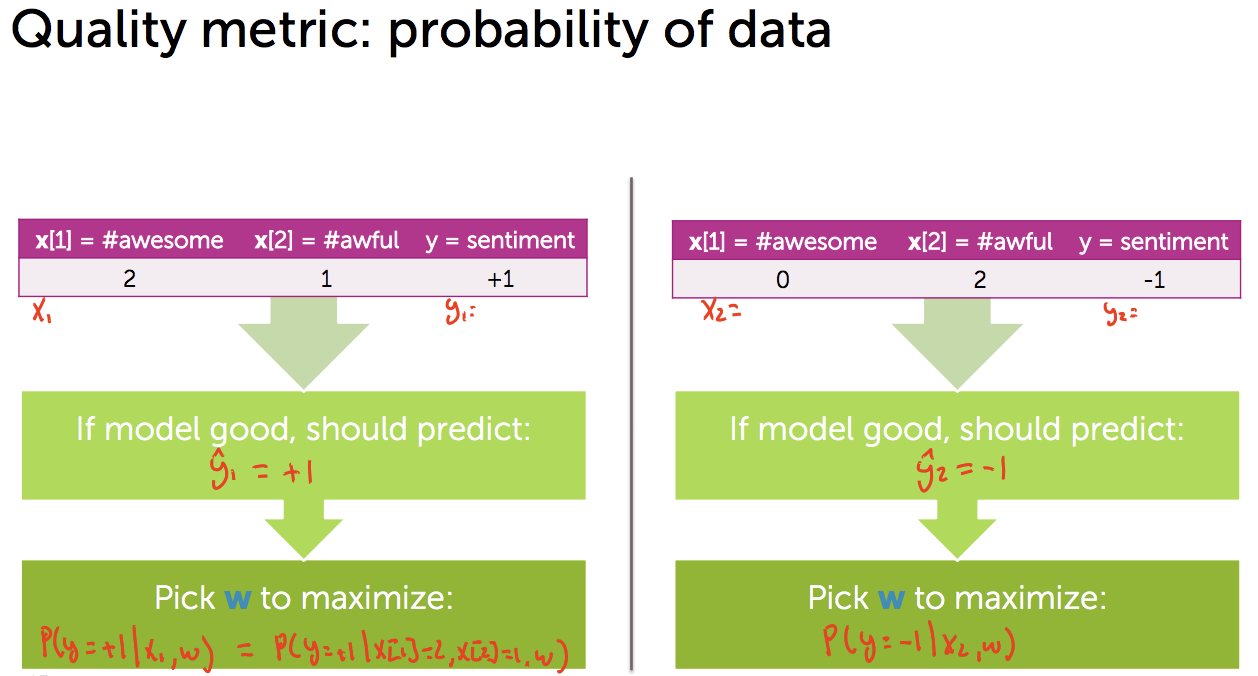
# How to Learn the parameters of logistic regression

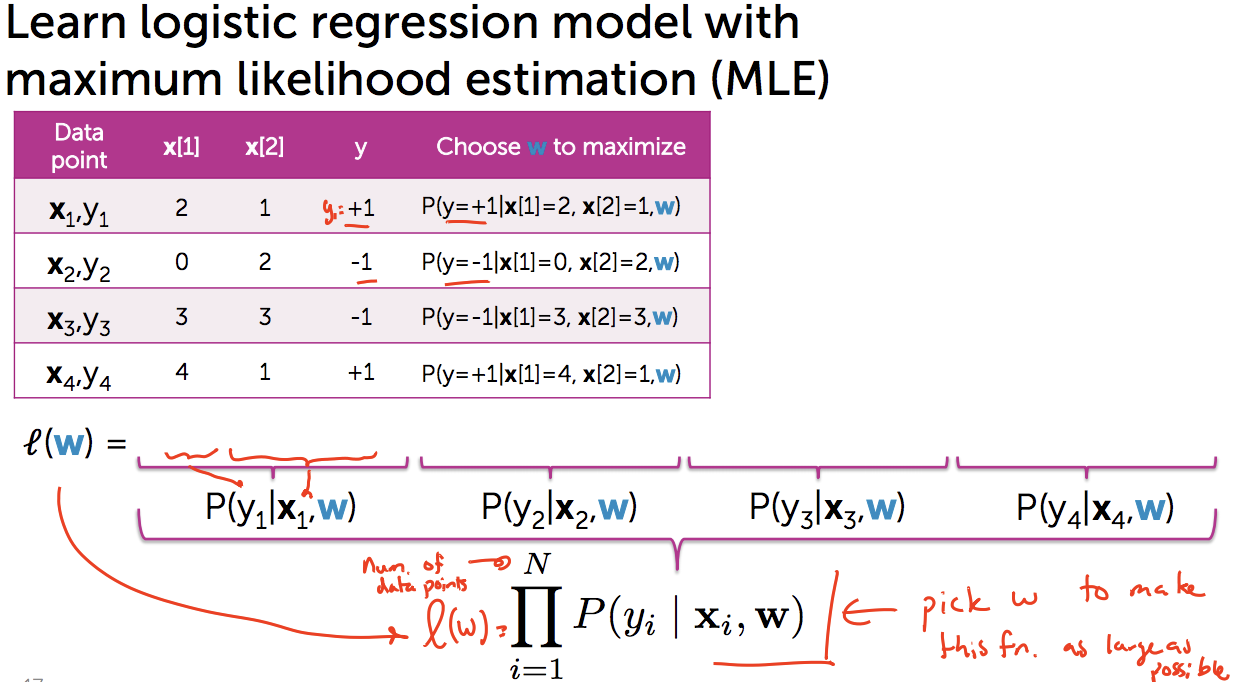
We will use training data to learn a **waight** or **coefficient** for each word.

Quality matric of logistic regression: Maximum likelihood estimation **(Data likelihood)**.

We need to find the coefficients (w) which will make the positive labled equation > 0.5 and negative labled equation < 0.5 and fit this attribute for all train data. Here we will use “gradient ascent” to do that.







We will use “Gradient Ascent” algorithm which will make the likelihood function value as big as possible. And thus we will find the best coefficients.