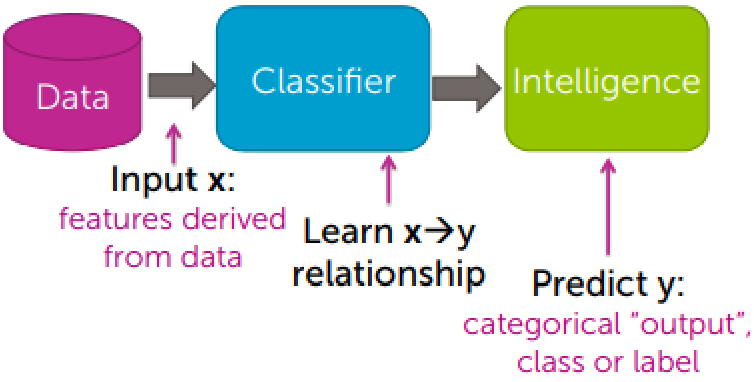
Classification

- From features to predictions.



Sentiment classifier:

- Takes a review -> classifier -> outputs whether - positive or negative review.

Multiclass classifier :

- Output has more than 2 categories.
- Input -> webpage;
- Output -> webpage content -> **Education** / **Finance** / **Technology**.

Spam Filtering:

- Input - Text of email, sender, IP address, etc;
- Output - **Not spam** / **Spam**.

Image Classification:

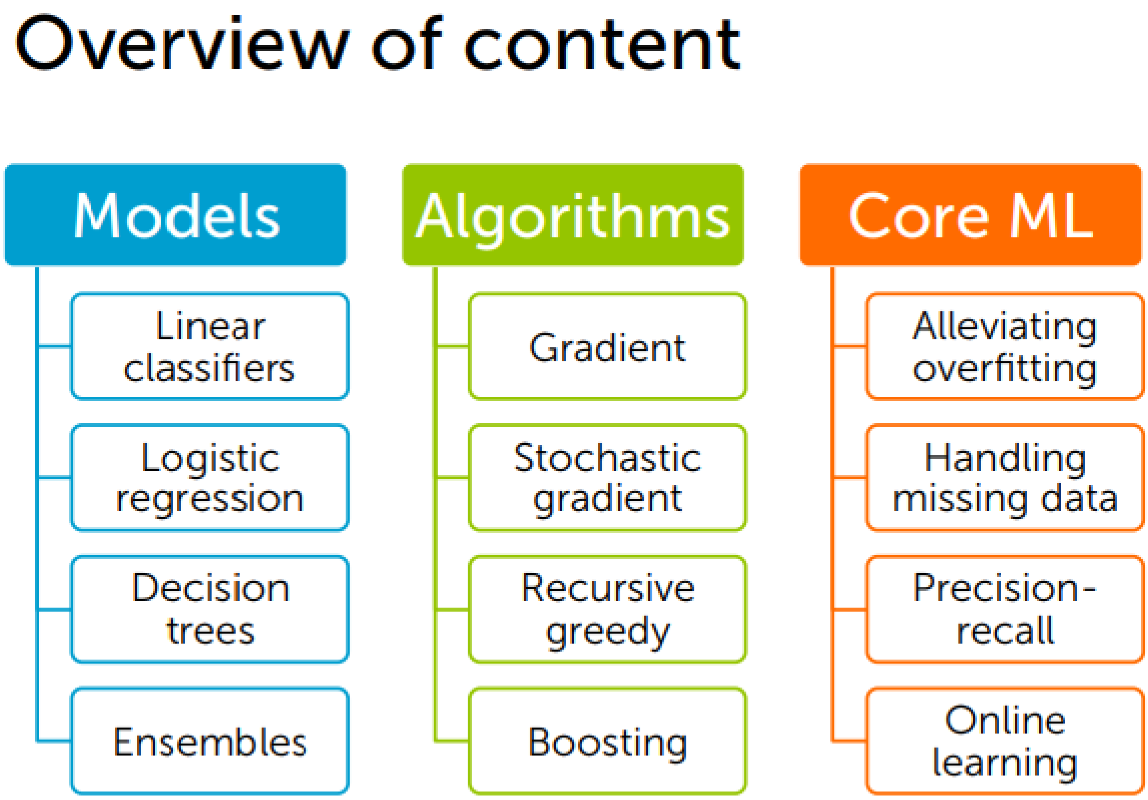
- Input - Image pixels
- Output - predicted object.

Personalized medicine:

- Input -> Thermometer temp, x-ray, lab-test, DNA sequence, lifestyle.
- Disease Classifier MODEL
- Output -> Healthy, Cold, Flu, etc.

Reading the mind:

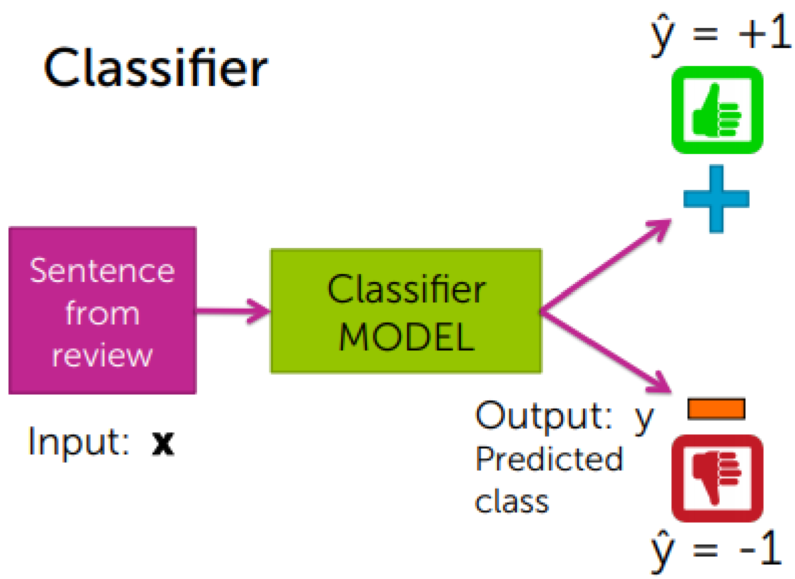
- Input - brain scan images - FMRI;
- Output - Image of the brain can predict what you are reading. (Hammer / House);



Linear Classifiers - Logistic Regression

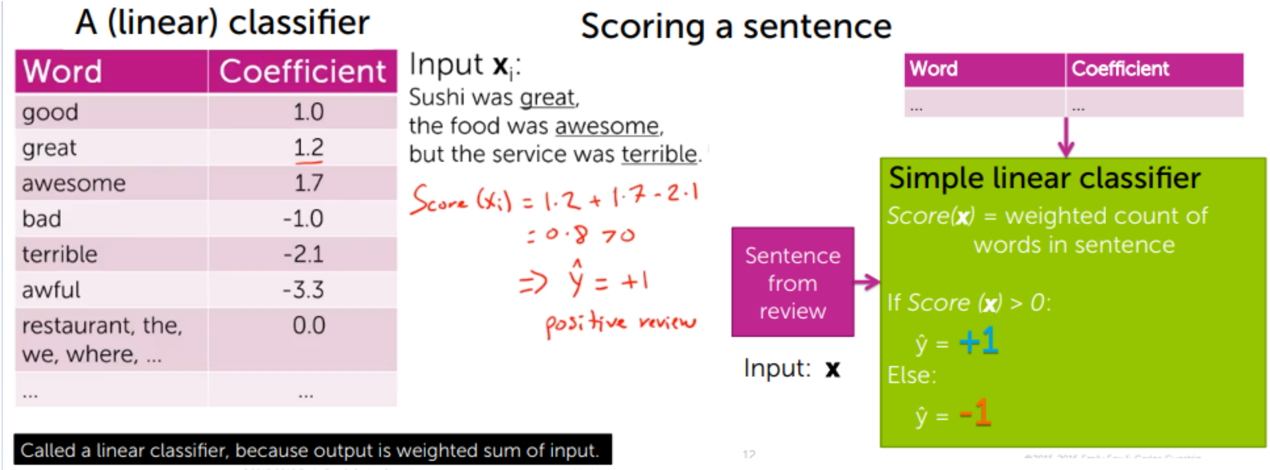
Classifier

- Take the input x -> feeds it to a classifer -> outputs prediction y-hat as positive / negative. (Equal weight); for 2 classes not multiclass.



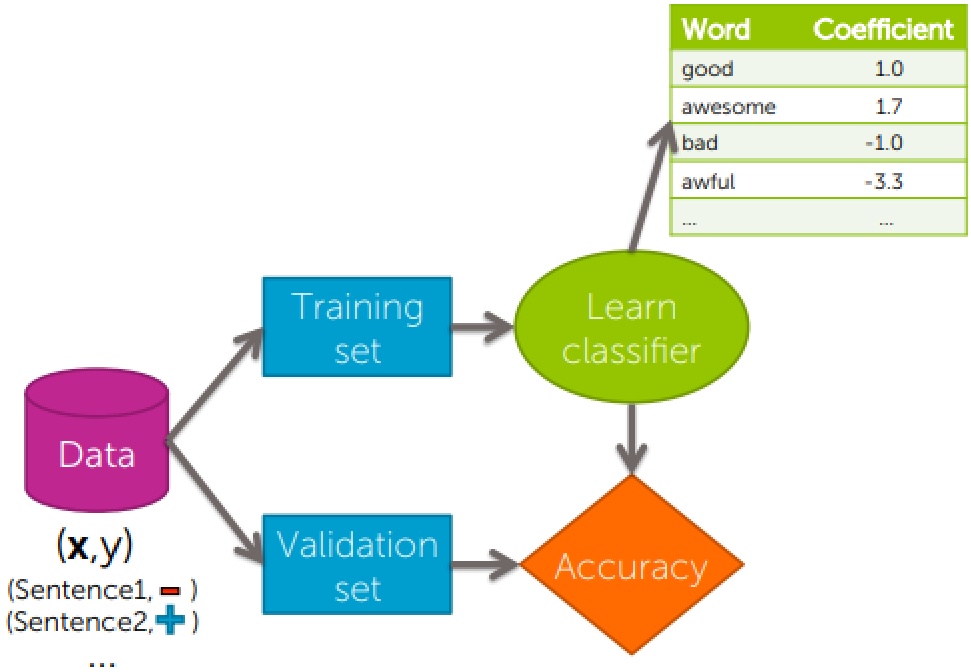
Linear classifier

- Uses the training data to learn the weights or coefficients for each word. Irrelevant words might have a score of 0.
- **Linear classifiers - output the weighted sum of the inputs**.



Training classifier -> Learning the coefficients

- Input dataset consists of reviews and their rating.
- The dataset is divided into **training set** and **validation set**.
- The training set is feed into a **learning classifier**. The learning classifier learns the weight associated with relevant words in the dataset.
- The **learning classifiers** accuracy is tested on the validation set.

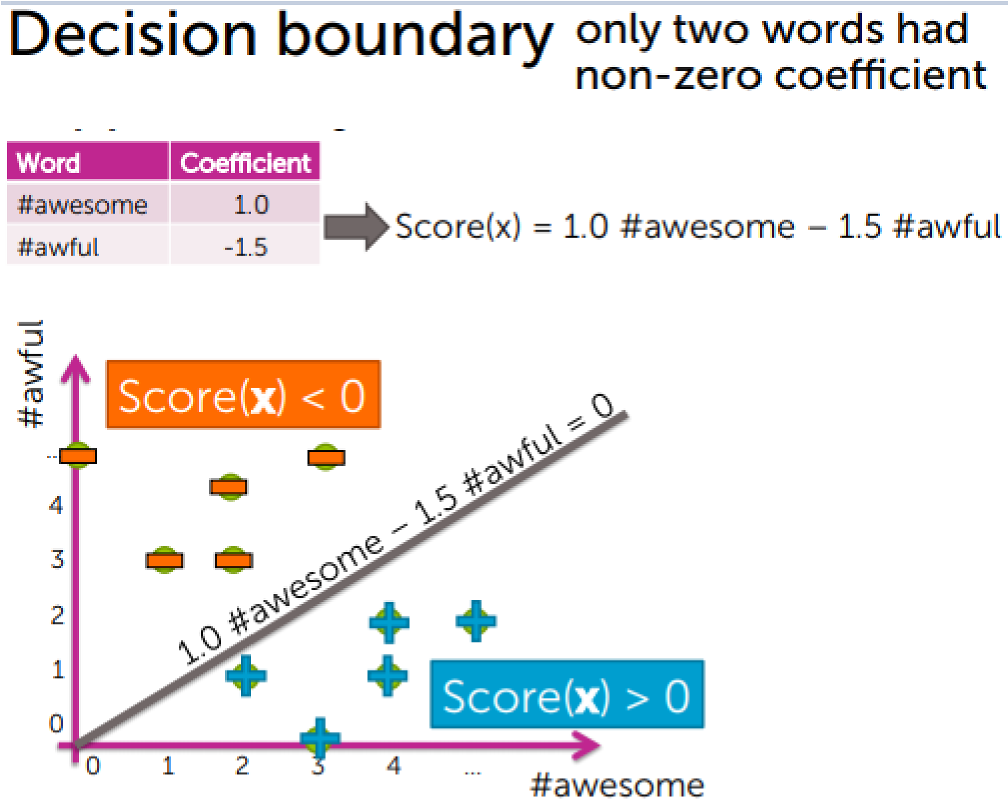


Decision boundaries

- It is the boundary between **positive** and **negative** predictions.

Consider two non-zero coefficients, and the rest are 0.

- Score(x) = 1.0 *#awesome -1.5* #awful;
- The dataset is plotted for it, and the **fit for this model is a line"**.
- Everything below the line is positive and every thing above the line is negative.



Word	Coefficient
#awesome	1.0
#awful	-1.5

Decision boundary separates positive and negative predictions:

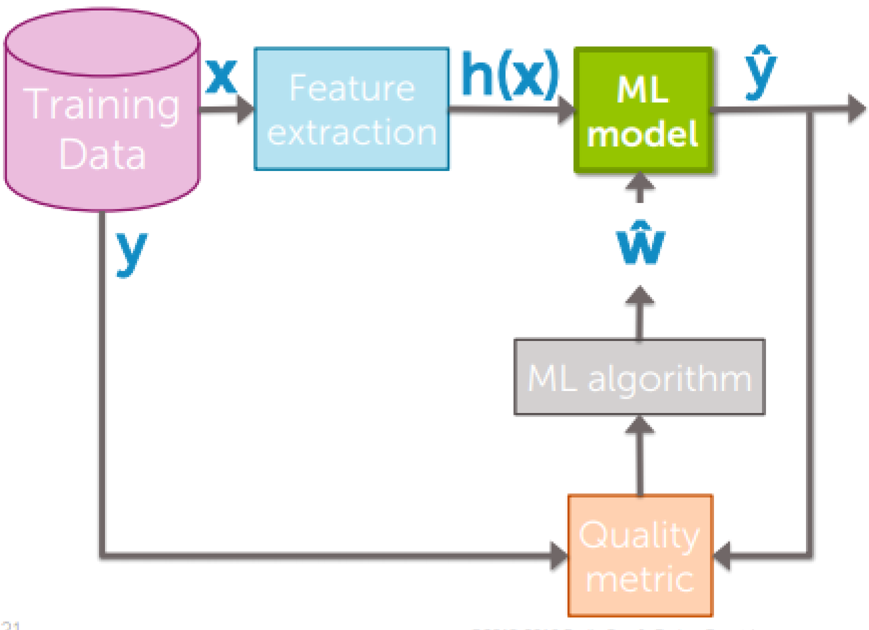
For linear classifiers -> (linear classifier - weighted sum of coefficients):

- When 2 coefficients are non-zero -> decision boundary - **line**.
- When 3 coefficients are non-zero -> decision boundary - **plane**
- When many coefficients are non-zero -> decision boundary - **hyperplane**.

For more general classifiers :

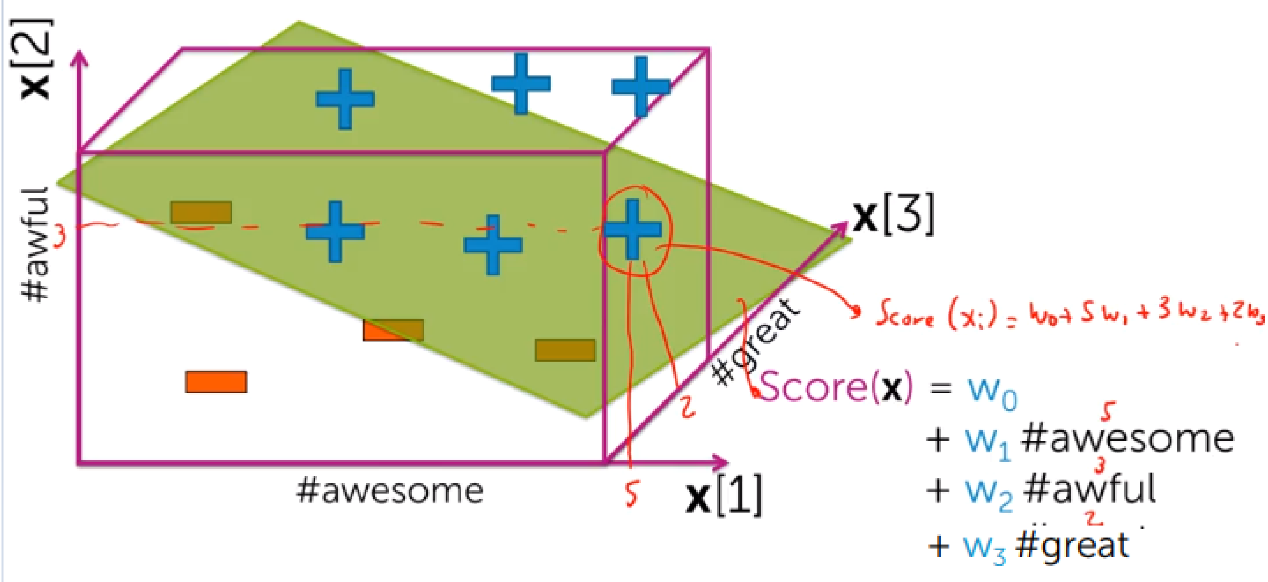
- more complicated shapes.

Linear classifier : Model



Coefficients of classifier

- Consider a 3-d space and a decision boundary built in that space.
- Based on the value of the coefficents the score can be classified as either positive or negative.



General notations

- Output : y -> {-1, +1}
- Inputs : x = {x[1],x[2],x[3],...,x[d]} -> x -> d-dimension vector.
- x[j] = jth input(scalar).
- hi(x) = jth feature (scalar).
- xi = input of ith data point (vector).
- xi[j] = jth input of ith data point (scalar).

Simple hyperplane

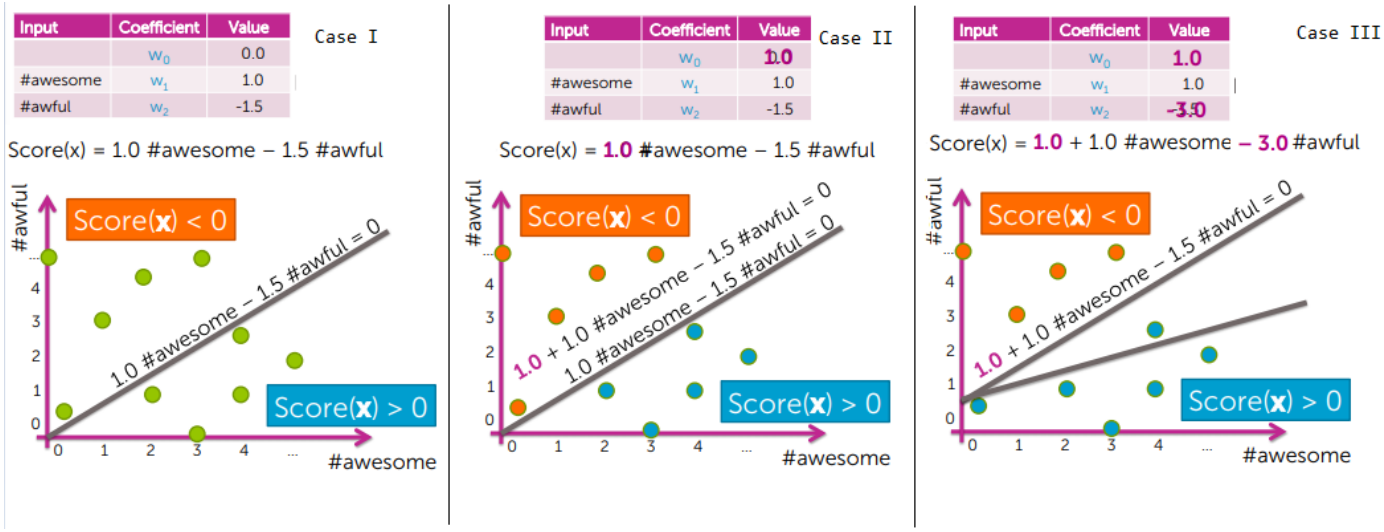
- **Model : y-hat = sign(Score(xi))**.
- **Score(xi) = w0 + w1 *xi[1]* + *w2* xi[2] + ... + wd *xi[d]* = *wT* xi;**
- The goal is to optimize wT *xi (w-transpose* xi);
  - feature 1 = 1
  - feature 2 = x[1] -> #awesome
  - feature 3 = x[2] -> # awful

    ...

  - feature d+1 = x[d] -> # ramen

Effect of 'Coefficient value' on 'Decision boundary'

- The coefficient value has a large impact on the decision boundary.
- Case I : w0 coefficient value = 0, the fit is a linear line that has n intercept 0 and hence starts at origin.
- Case II : w0 coefficient value = 1, the fit is linear but shift since the intercept in 1, and certain data-points in the negative review side become positive.
- Case III : Thw w2 coefficient value decreases from -1.5 to -3 (magnitude increases) - the fit line becomes less steep and hence a few of the data-points from positive review become negative.



Model in terms of features h(x) rather that inputs x

Model: $\hat{y}_i = \text{sign}(\text{Score}(\mathbf{x}_i))$

$$\text{Score}(\mathbf{x}_i) = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i)$$

$$= \sum_{j=0}^{D} w_j\, h_j(\mathbf{x}_i) = \mathbf{w}^T h(\mathbf{x}_i)$$

feature 1 = $h_0(\mathbf{x})$ ... e.g., 1

feature 2 = $h_1(\mathbf{x})$ ... e.g., $\mathbf{x}[1]$ = #awesome

feature 3 = $h_2(\mathbf{x})$ ... e.g., $\mathbf{x}[2]$ = #awful

or, log($\mathbf{x}$[7]) $\mathbf{x}$[2] = log(#bad) x #awful

or, tf-idf("awful")

...

feature D+1 = $h_D(\mathbf{x})$ ... some other function of $\mathbf{x}[1], \ldots, \mathbf{x}[d]$

Class Probabilities

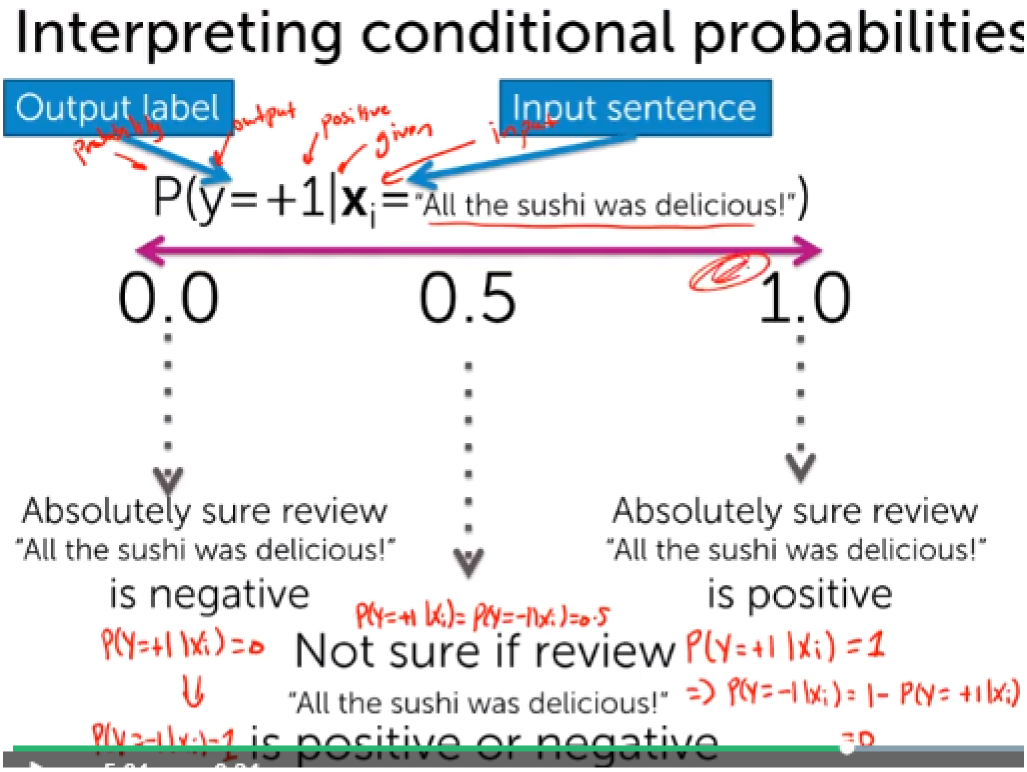
- The prediction may not always be a definite potive or definite negative one.
- Some reviews may be uncertain. between postive P(1) and negative P(0).
- P(0) <= x <= P(1);

Key properties of probabilities

- Probability always lies between 0 & 1. [0,1]
- Probabilities sum up to 1. P(1) + ... + P(d) = 1

Conditional probability

- It is the probability of the output given the input is true.



Key properties of conditional probabilites'

- Conditional probabilities always between 0 & 1.
  - 0 <= P(y=+1 | xi) <=1
  - 0 <= P(y=-1 | xi) <=1

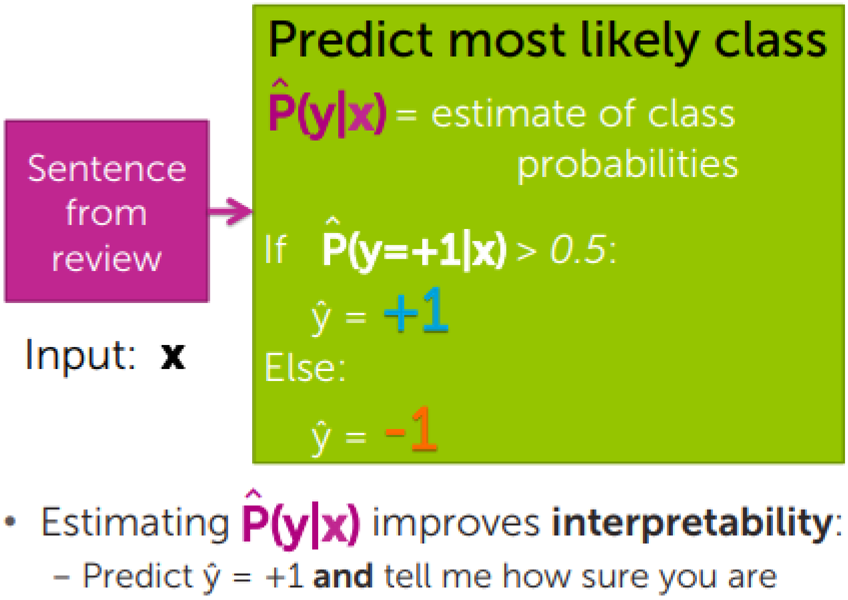
  - 0 <= P(y=dog | xi) <=1
  - 0 <= P(y=cat | xi) <=1
  - 0 <= P(y=bird | xi) <=1
- Conditional probabilities sum up to 1 over y, but not over **x**.
  - P(y=+1 | xi) + P(y=-1 | xi) = 1

  - P(y=dog | xi) + P(y=cat | xi) + P(y=bird | xi) = 1

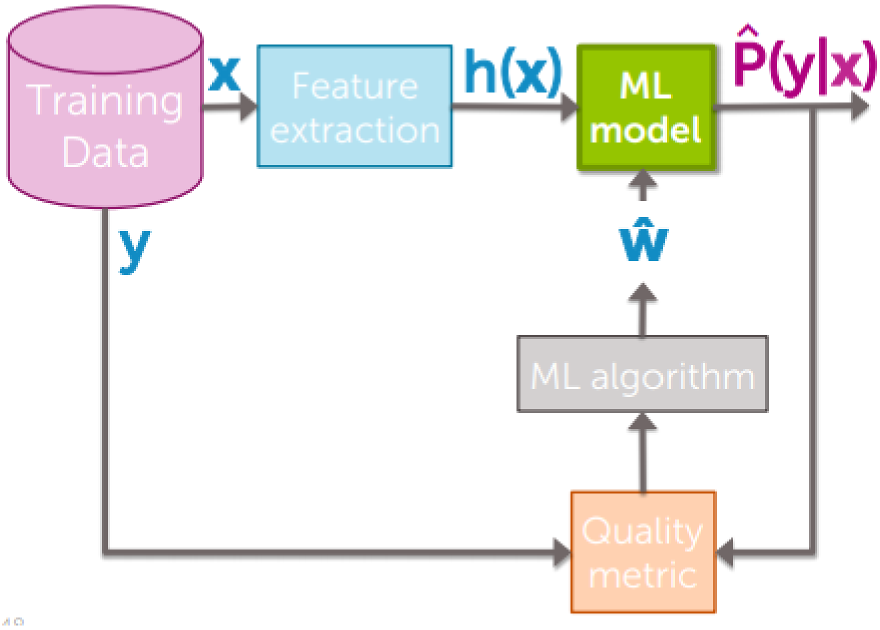
Prediciton confidence

- The conditional probability -> improves the degree of certainty of the data.
- **P(y|x)** -> y - output label; x -> input label;
- Sentence 1 - x - "The sushi and everything is awesome!";
  - P(y = +1 | x) = 0.99;
- Sentence 2 - x = "The sushi was good, the service was OK";
  - P(y = +1 | x) = 0.55;

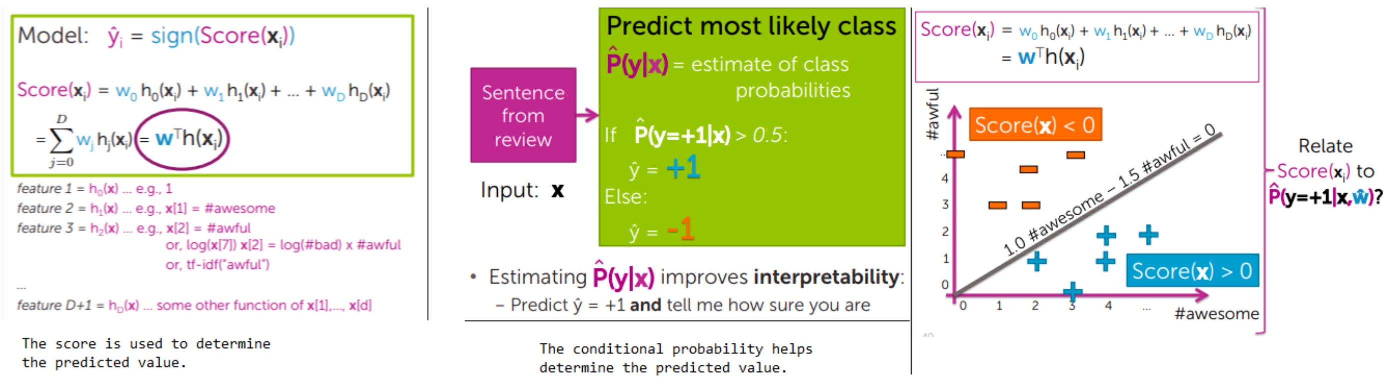
Goal is to learn conditional probabilities from the data



The ML Block diagram



The decision boundary helps segregate the positive and negative reviews.



- Need to relate **Score(xi)** to **P-hat(y=+1|x, w-hat)**.

Relating Score(xi) to Probability

- **The Score(xi) -> range [-infinity, +infinity]**.
- **The range of output estimate y-hat = [-1,+1]**.
- **The probability -> range [0,1]**.
  - The output estimate y-hat -> +1 if Score(xi) > 0;
  - The output estimate y-hat -> -1 if Score(xi) < 0;
  - The probability P(y=+1|xi) = 1, if ***y-hat = 1 with certainity***.
  - The probability P(y=+1|xi) = 0, if ***y-hat = -1 with certainity***.
  - The probability P(y=+1|xi) = 0.5, if ***y-hat is -1 or +1 | not sure***.

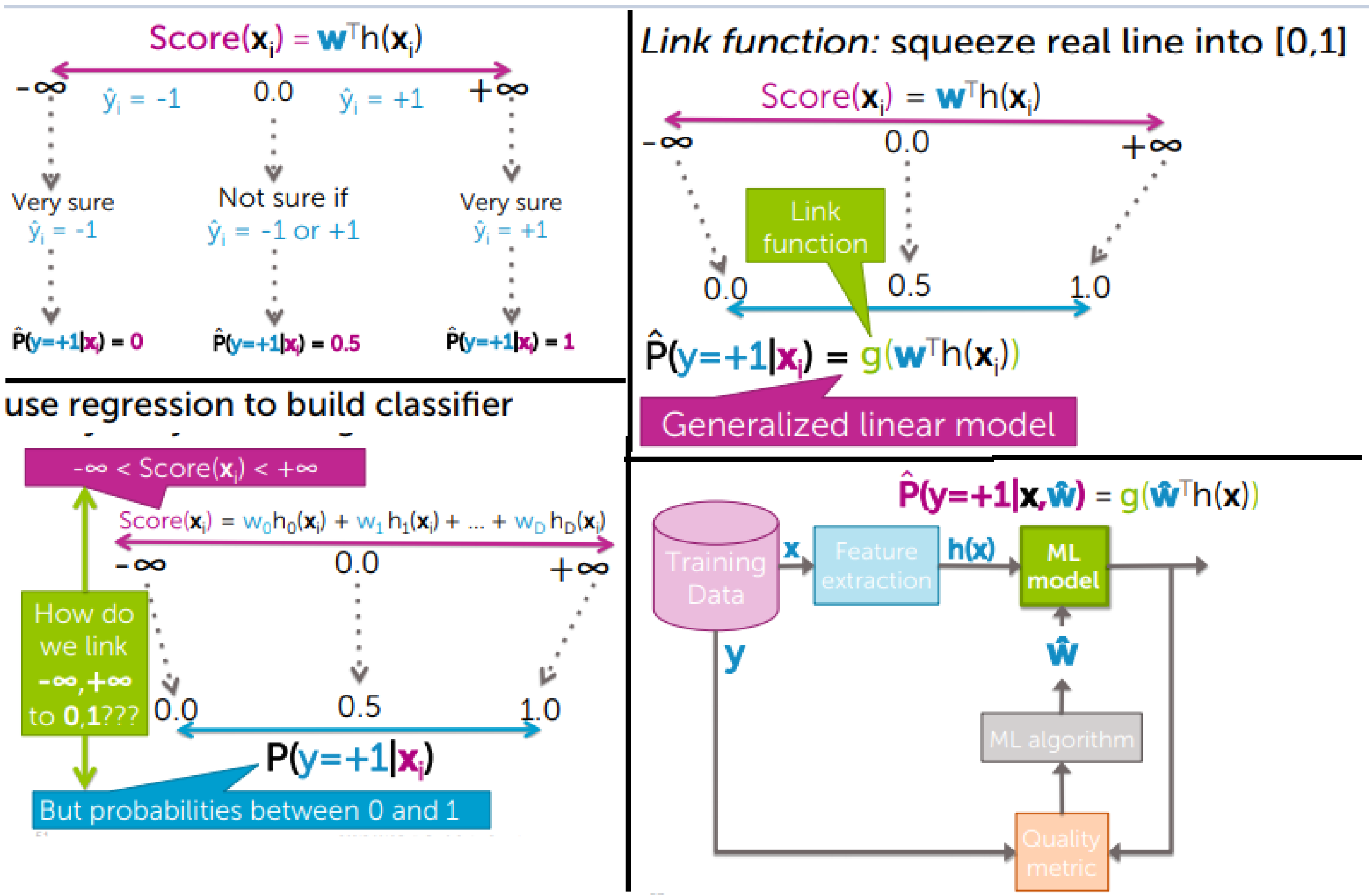
Using regression to build classifier since - The Score(xi) is the sum of the weights of the features.

- The score that ranges from -infinity to + infinity should be reduced to conditional probability that ranges from 0 to 1.
- This can be achieved through **link functions**.
- The **Link function** squeezes the real line into [0,1].

Generalized Linear model

- It is a link function that raduces a real line [-infinity, +infinity] to intervals[0,1].
- There certain generalized linear classifiers taht don't squeeze to [0,1];

The goal -> Taking the data from feature extraction (TFIDF, etc) -> Build a linear model (w-transpose * h(x)) -> Push it through the link function that squeezes it into interval [0,1] -> Use that to predict the probability of the sentiment, given the input sentence.



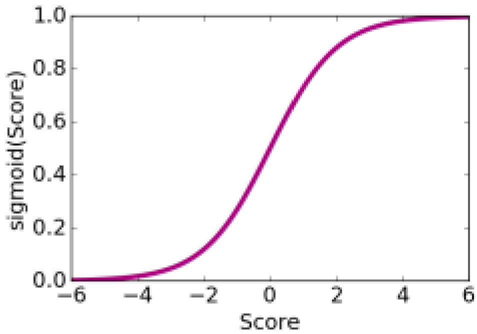
Sigmoid (or logistic) link function

- Logistic Regression -> Specific case of generalized linear classifier, where the logistic function is used to squeeze the real data range [-infinity, + infinity] to [0,1] range inorder to predict probability for every class.
- **The 'Logistic Regression' uses Logistic function / sigmoid / logit**.
- sigmoid(Score) = 1 / (1 + e^-Score);
- The sigmoid function ranges from [0,1] for input range [-infinity, +infinity];
  - Score - x = -infinity | sigmoid(Score) = 0;
  - Score - x = +infinity | sigmoid(Score) = 1;
  - Score - x = 0 | sigmoid(Score) = 0.5;

Logistic function (sigmoid, logit)

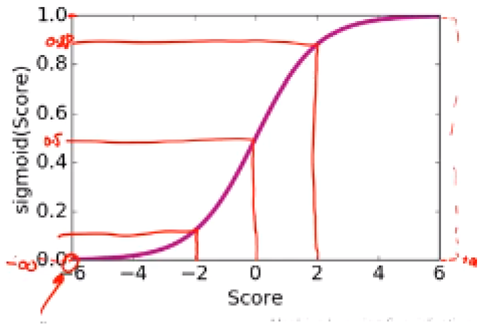
$$sigmoid(\text{Score}) = \frac{1}{1+e^{-\text{Score}}}$$

Score	$-\infty$	-2	0.0	+2	$+\infty$
sigmoid(Score)					

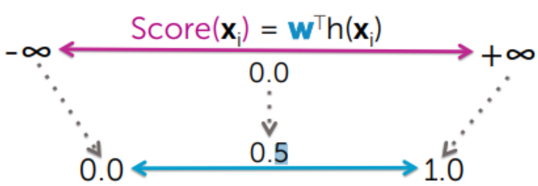


Score	$-\infty$	-2	0.0	+2	$+\infty$
sigmoid(Score)	$\frac{1}{1+e^{\infty}} = \frac{1}{1+\infty} = 0$	0.12	Sigmoid(0) $= \frac{1}{1+e^{0}} = \frac{1}{1+1} = 0.5$	0.88	$\frac{1}{1+e^{-\infty}} = 1$

$e^{\infty} = \infty$ $e^{0} = 1$ $e^{-\infty} = 0$



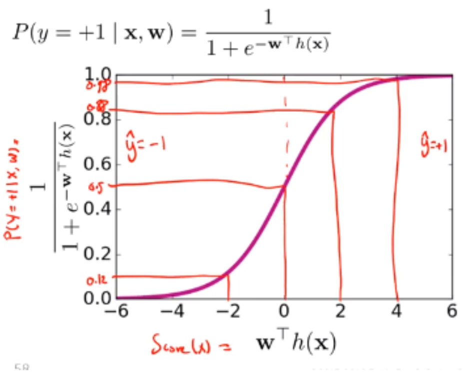
Logistic regression model



P(y=+1|$\mathbf{x}_i$,$\mathbf{w}$) = sigmoid(Score($\mathbf{x}_i$))

$= \frac{1}{1+e^{-Score(x_i)}} = \frac{1}{1+e^{-w^T h(x_i)}}$

$= \frac{1}{1+e^{-(w_0 h_0(x_i) + w_1 h_1(x_i) + \dots + w_D h_D(x_i))}}$

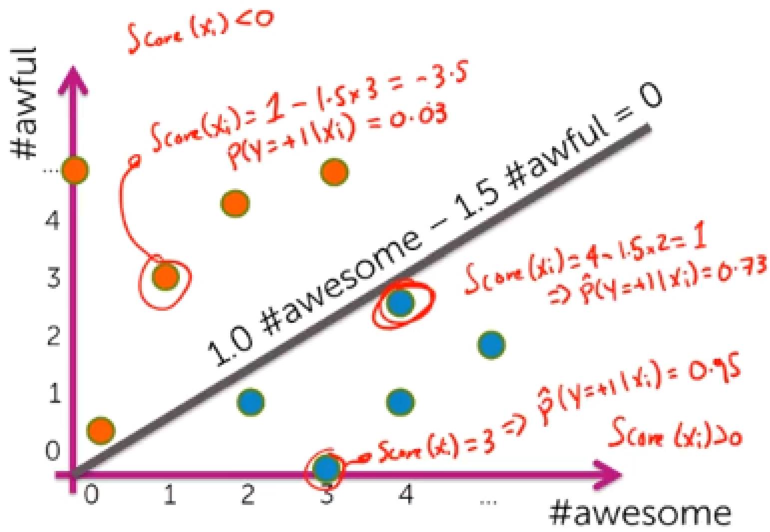


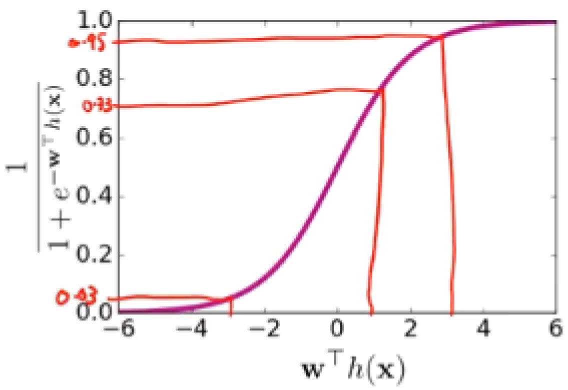
Score($\mathbf{x}_i$)	P(y=+1|$\mathbf{x}_i$,$\mathbf{w}$)
0	0.5
-2	0.12 < 0.5 ⇒ $\hat{y}=-1$
2	0.88 ⇒ $\hat{y}=+1$
4	0.98 ⇒ $\hat{y}=+1$

```
If P(y-hat=+1 |xi, w) > 0.5 :
        y-hat = +1
else
        y-hat = -1
```

Effects of coefficient values on predicted probabilities

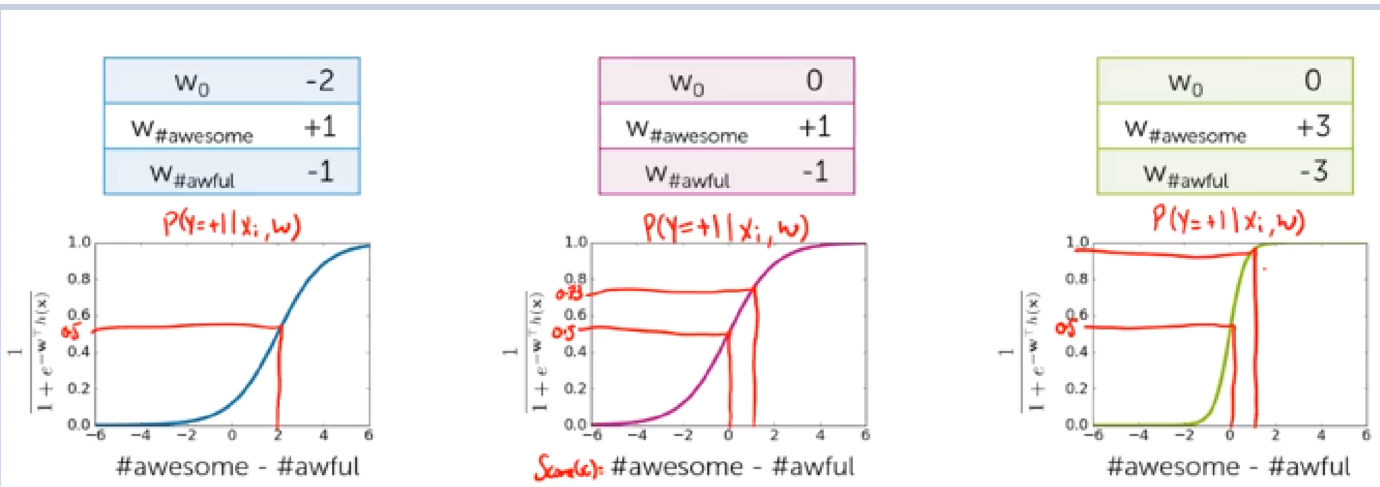
- From the logistic regression -> linear decision boundary.
- The points below the line are positive, while those above the line are negative.
- The line or the decision boundary has a **Probability = 0.5** and **Score(xi) = 0**.
- As the datapoints get further away from the boundary their certainity increases accord to the sigmoid curve.





The sigmoid curve varies with the value of the coefficients.

- Case I : w0 = -2, w1 = + 1, w2 = -1
  - Score = #awesome - #awful = 2; P = 0.5;
- Case II : w0 = 0, w1 = + 1, w2 = -1
  - Score = #awesome - #awful = 0; P = 0.5;
- Case III : w0 = 0, w1 = +3, w2 = -3 | Higher magnitude - steeper curve.
  - Score = #awesome - #awful = 0; P = 0.5;



As the magnitude of the parameters increasesthe prediction certainity based on the probability is achieved faster.

Overview Learning Logistic Regression Models

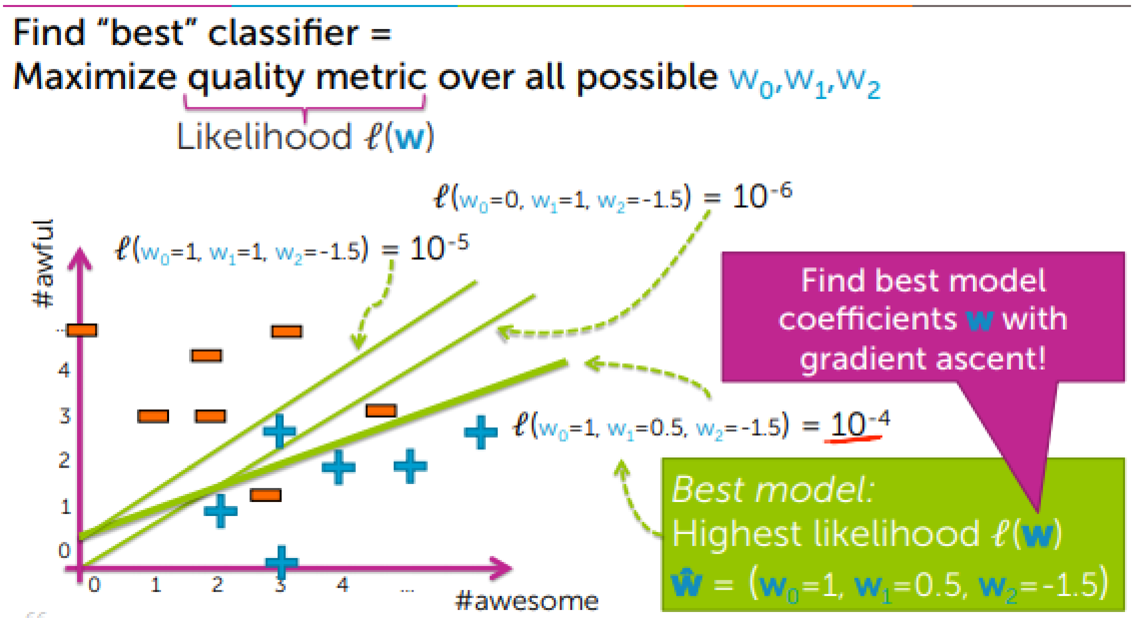
- The Dataset is split into **Training set** and **Validation Set**.
- From the **training set** run a learning algorithm that outputs the **parameter estimates w-hat**(coefficient values -> good = 1.0, awful = -3.3);

- The **w-hat** are plugged into the model - inorder to estimate the probability of a sentence - whether it is positive or negative.
- The learned model can be used to estimate on the validation set. Quality metrics.

Quality metrics

- Inorder to find the best classifier - **maximize the quality metrics - (Likelihood l(w)) - over all possible coefficients - w0,w1,w2**.

The 'Gradient Ascent' algorithm is used to find the set of parameters w that has the 'Likelihood' best quality.



Practical issues with classifications

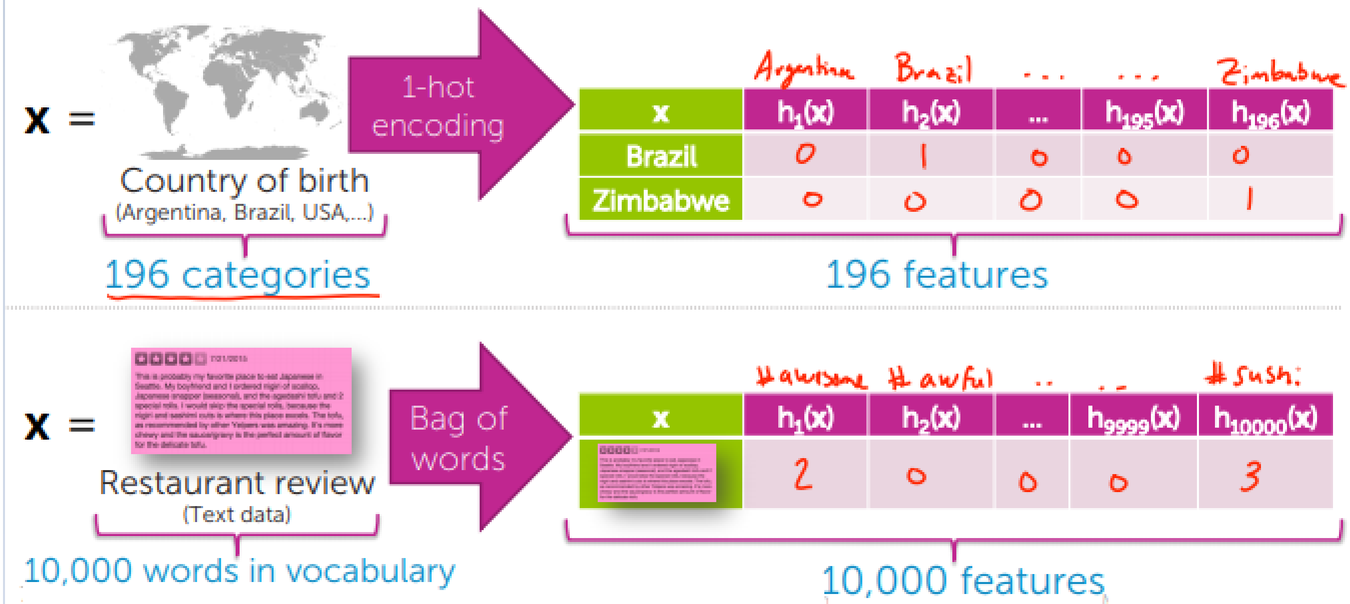
Encoding categorical inputs:

- The input data thus far has been **numerical data**.
  - #awesome, #age, #salary;
  - These features can be multipled by the coefficients and will be intuitive.
- Categorical inputs: These are non-numberical inputs and don't necessarily increase with scale.
  - Gender, Country, zipcode.
  - Example Zipcode - 10050 and 98654 -> there are different zipcodes to locate regions and the numerical value of the zipcode does not make the higer zipcode more valued. Hence it cannot be considered as numerical value and the result will not be intuitive when multiplied by a coefficient.

Therefore must convert categorical inputs into numeric features.

Encoding categories as numeric features

- 1-hot encoding : Take the input x (country of birth) ,and 1-Hot encoding that creates one feature for every possible country. Example - h1(x) - Argentina, h2(x) - Brazil, ... hd(x) - Zimbabwe;
  - Here only one of features will have the value 1 and the rest will be 0.
- Bag-of-words encoding : Take input x (restaurant review -> 10,000 words in vocabulary), Bag-of-words takes the text and then codes it as a count. Each of the features is the count of words. h1(x) - #awesome, h2(x) - #awful,...

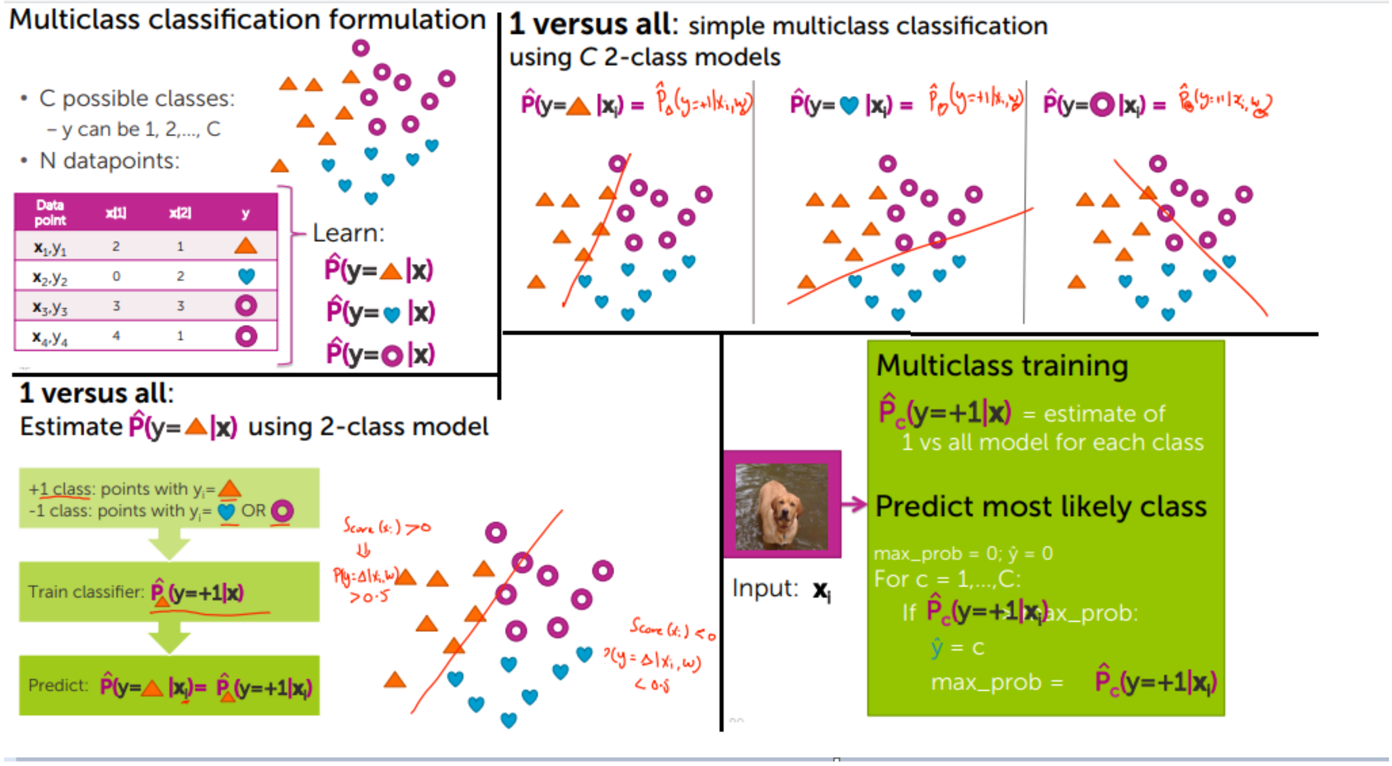


In the above two cases the categorical input data is taken and defined a set of features one for each possible category to contain either a single value or account and feed it into the 'logistic regression' model.

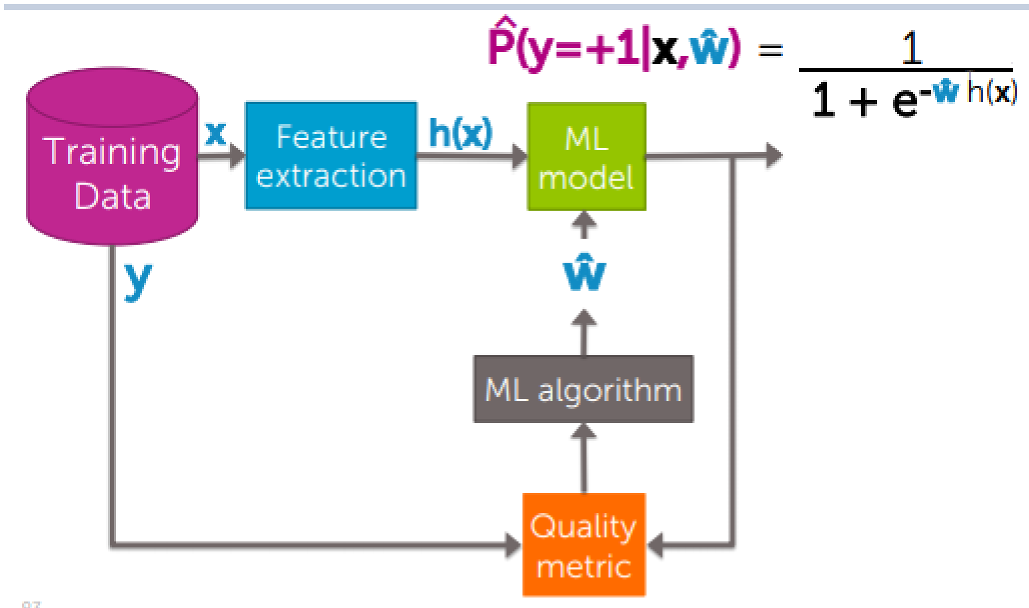
Multiclass classification with 1 versus all

- The input in and image and the output is to be predicted. Here the output is not just -1 or +1. The range of the output can vary.
- There are many approaches to solve such scenarios.
- **1 versus all**
  - There can be 'C' possible classes (y =1,2,3,..,.,C);
  - N datapoints -> labelled as in the datapoint will be associated with y (class);
  - Need to learn, the probability of the output belonging to a particular class given the input.
  - For each observation it is checked through all classes, the class that outputs the highest probability will be the probability class for that observation.
- Consider 2 classes -> triangles, hearts and donuts,
  - The datapoints are labelled -> input and y the class in provided for the dataset.
  - In "1 versus all" approach - need to estimate the probability that y is a particular class given x.
- Example: Estimate y is a triangle given the input x:
  - Here output y-hat = +1 for points with labbel input yi - traingle and -1 for others (hearts / donuts);
  - The yi= triangle observations are passes through to train the classifier for the class.
  - The prediction is performed.

this is performed across all classes. For each input xi, the based on the yi and the class that has the maximum probability based on yi the max-probability is assigned to that classs - it has y-hat = +1 and rest of the classes are -1.



Logistic regression model



likelihood

Quiz

1.

(True/False) A linear classifier assigns the predicted class based on the sign of $\text{Score}(x) = \mathbf{w}^T h(\mathbf{x})$.

- (selected) True
- False

2.

(True/False) For a conditional probability distribution over $y|\mathbf{x}$, where y takes on two values (+1, -1, i.e. good review, bad review) $P(y = +1|\mathbf{x}) + P(y = -1|\mathbf{x}) = 1$.

- (selected) True
- False

3.

Which function does logistic regression use to "squeeze" the real line to [0, 1]?

- (selected) Logistic function
- Absolute value function
- Zero function

4.

If $\text{Score}(x) = \mathbf{w}^T h(\mathbf{x}) > 0$, which of the following is true about $P(y = +1|\mathbf{x})$?

- P(y = +1 | x) <= 0.5
- (selected) P(y = +1 | x) > 0.5
- Can't say anything about P(y = +1 | x)

5.

Consider training a 1 vs. all multiclass classifier for the problem of digit recognition using logistic regression. There are 10 digits, thus there are 10 classes. How many logistic regression classifiers will we have to train?

10