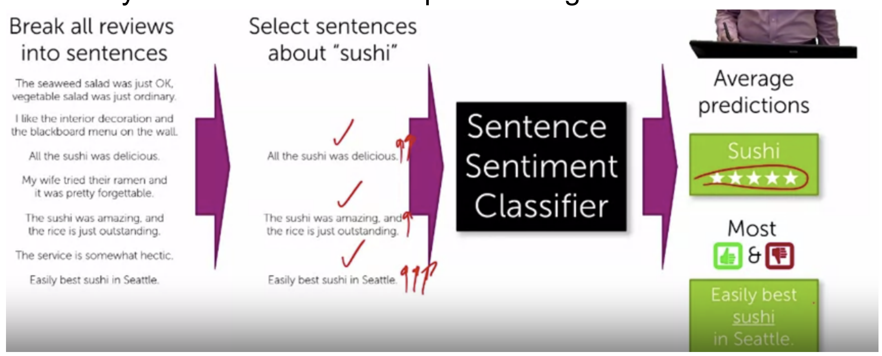
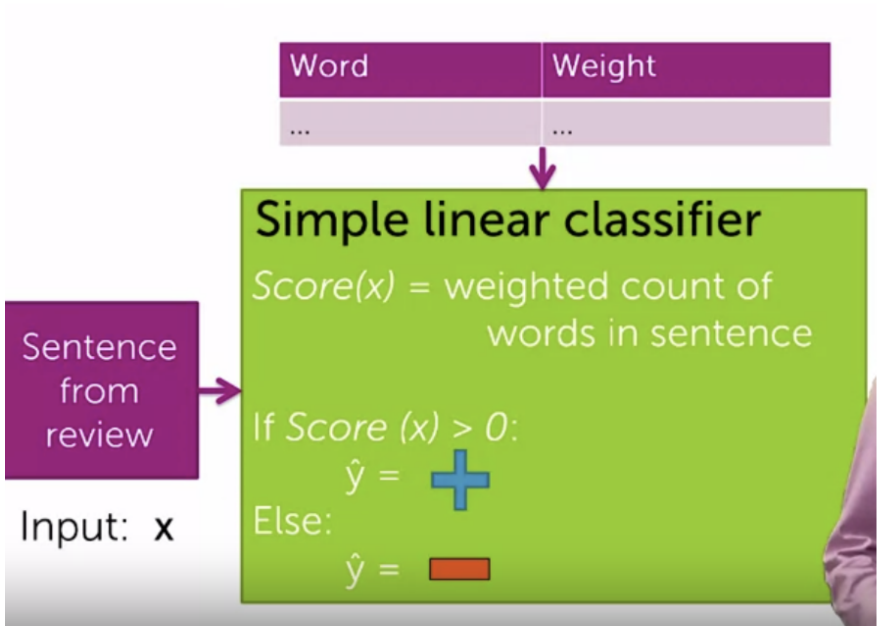
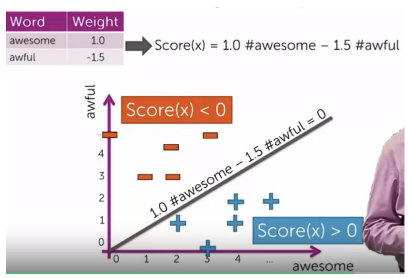
<table>
<tr><th colspan="2">Classification Modeling</th></tr>
<tr><td>What is Classification</td><td>A supervised ML method where the model tries to predict the correct label of a given input data. The model is fully trained using the training data and then it is evaluated on test data before being used on new unseen data.</td></tr>
<tr><td>Examples</td><td>Sushi restaurant review. Take the reviews, break down all the sentences, and use a “sentence sentiment classifier” to average how many of the sentences had positive/negative sushi reviews.



</td></tr>
<tr><td>Examples</td><td>Ex: Webpage
- Input: webpage
- Output: classifier of education, finance, technology
Ex: Spam filtering
- Input: Text of email, sender, IP….
- Output: Spam/Not Spam
Ex: Image Classification
- Input: Image pixels
- Output: Predicted object
Ex: Personalized medical diagnosis
- Input: symptoms/measurements
- Output: Disease diagnosis</td></tr>
<tr><td>Linear Classifiers</td><td>Use an object’s characteristics to identify which class or groups it belongs to.

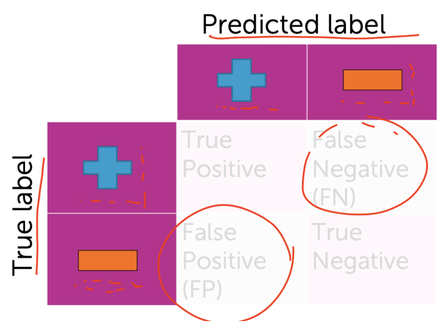
Ex: To determine if a restaurant review is positive or negative, we add weight to each word (addresses the issue of which words weigh more than others) and have a list of those words. We can address “good vs not good” from a more complex algorithm.</td></tr>
</table>

<table>
<tr><td></td><td>
</td></tr>
<tr><td>Decision Boundaries (How linear classifiers make decisions)</td><td>Decision boundaries are what separates positive and negative predictions.
- 

Visualizing decision boundaries and linear classifiers:
- 2 weights => linear
- 3 weights => plane
- 4+ weights => hyper plane
- More general classifiers => more complicated shapes</td></tr>
<tr><td>Training and Evaluating a Classifier</td><td>First train the classifier, then gauge the classification error and accuracy from the test set to determine how correct our model is and how to learn from it.

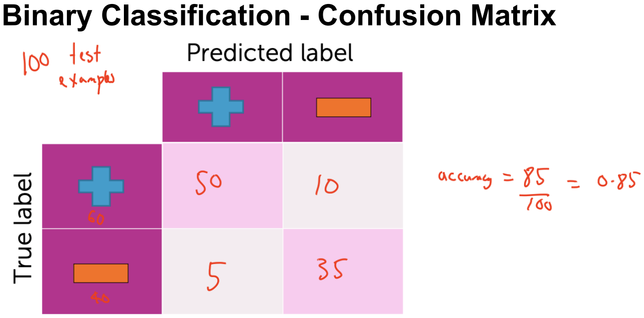
Error measures fraction of mistakes, best possible value is 0
$$Error = \frac{\#\ of\ Mistakes}{\#\ of\ Total\ Items}$$
Accuracy measures fraction of correct predictions, best possible v value is 1
$$Accuracy = \frac{\#\ Correct}{\#\ of\ Total\ Items}$$</td></tr>
</table>

<table>
<tr><td>Accuracy</td><td>
At a minimum, classifier should beat random guessing (50% for binary items, 25% for items with 4 choices)

Is a classifier with 90% accuracy good? Not necessarily
- Majority class prediction, where the majority is already right. Example is 90% of emails are already spam, so a spam classifier is bound to have higher chances of accuracy
</td></tr>
<tr><td>Types of Mistakes</td><td>
False Positives
- Should actually be negative. Type 2 error
False Negatives
- Should actually be positive. Type 1 error
Confusion Matrices


</td></tr>
<tr><td>Types of Mistakes Example</td><td>
Ex: Spam
- FN: Spam email failed to be labeled as spam
- FP: non-spam email falsely identified as spam
Ex: Medicine
- FN: Has disease, was not identified
- FP: Does not have disease, falsely identified as sick

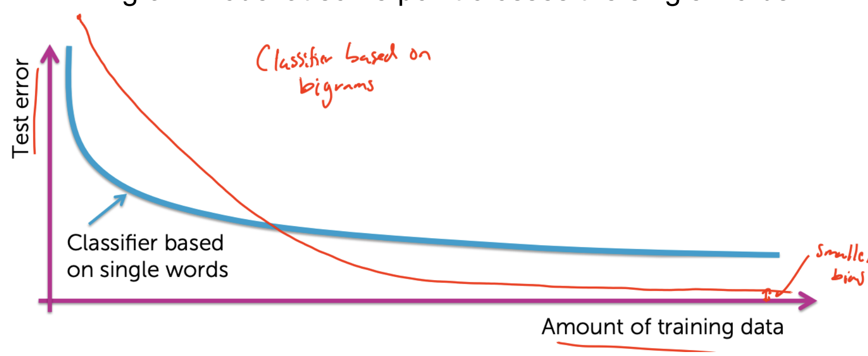
<table>
<tr><th></th><th>Spam filtering</th><th>Medical diagnosis</th></tr>
<tr><td>False negative</td><td>Annoying</td><td>Disease not treated</td></tr>
<tr><td>False positive</td><td>Email lost
Higher cost</td><td>Wasteful treatment</td></tr>
</table>

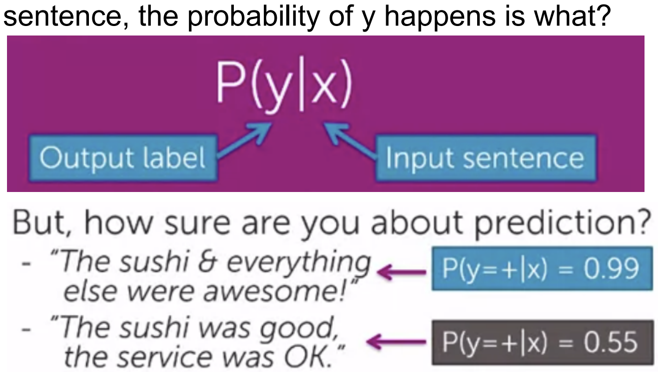
Binary Classification - Confusion Matrix


</td></tr>
</table>

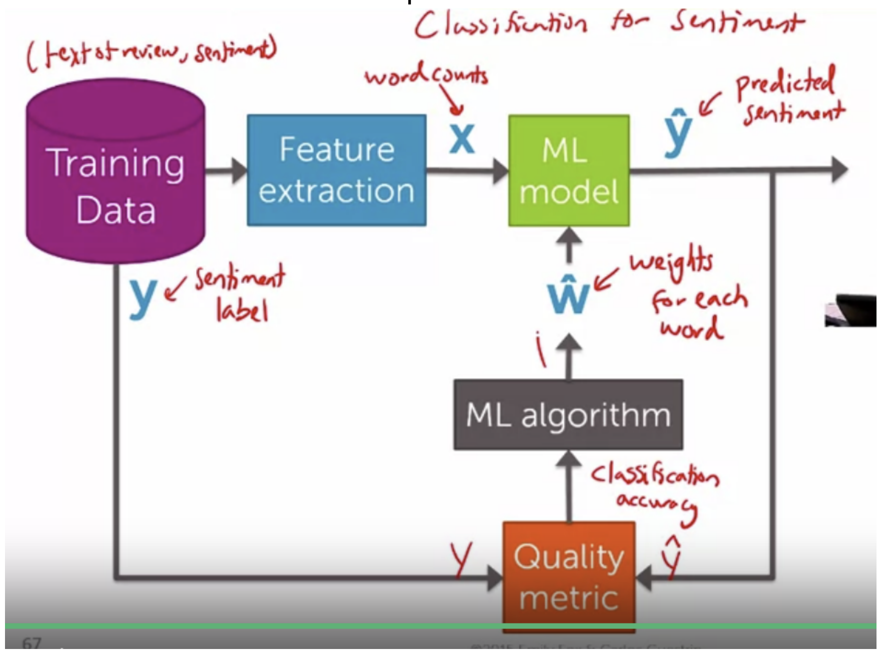
<table>
<tr>
<td></td>
<td>
Multiclass Classification - Confusion Matrix
<table>
<tr><td>100 test examples</td><td colspan="3">Predicted label</td></tr>
<tr><td>True label</td><td>Healthy</td><td>Cold</td><td>Flu</td></tr>
<tr><td>Healthy 70</td><td>60</td><td>8</td><td>2</td></tr>
<tr><td>Cold 20</td><td>4</td><td>12</td><td>4</td></tr>
<tr><td>Flu 10</td><td>0</td><td>2</td><td>8</td></tr>
</table>
accuracy $= \frac{80}{100} = 0.8$
</td>
</tr>
<tr>
<td>Learning Curves
How much data does a model need to learn?</td>
<td>
The more data the better, but quality over quantity. Theoretical techniques only provide guidance. In practical application, complex models require more data.

Even with infinite data, the best data will not go to zero test error.

Models with more parameters will get less errors than models with fewer parameters, but will need more data
- Ex) Bigram models can classify “no good” but require more data than single word models trying to classify “good”. Bigram model at some point crosses the single words.


</td>
</tr>
<tr>
<td>Class Probabilities
How confident is your prediction?</td>
<td>
Many classifiers provide a confidence level: Given the input sentence, the probability of y happens is what?


</td>
</tr>
</table>

<table>
<tr>
<td>Classification ML Block Diagram</td>
<td>Flow of ML Block Diagram
1. Feed in Training Data such as text of review, sentiment
2. Extract the details to be analyzed (such as word count)
3. ML Model takes in the words (x) and the weights for each word (w-hat) and outputs the predicted sentiment (p-hat)
4. The Quality Metric then compares the predicted output (p-hat) with the actual sentiment (y) and finds the classification accuracy
5. ML Algorithm updates the weights (w-hat) for each word to refine the next round of predictions


</td>
</tr>
</table>