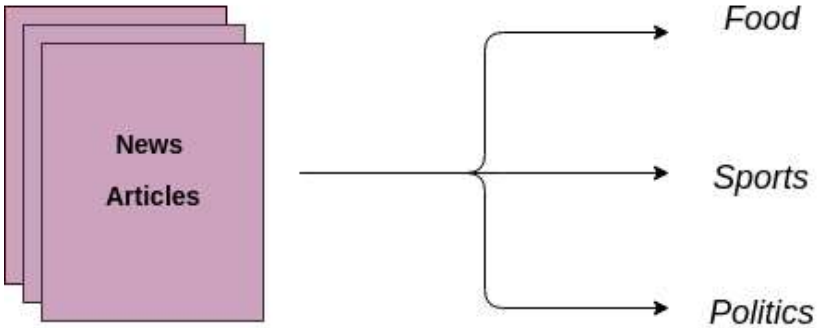
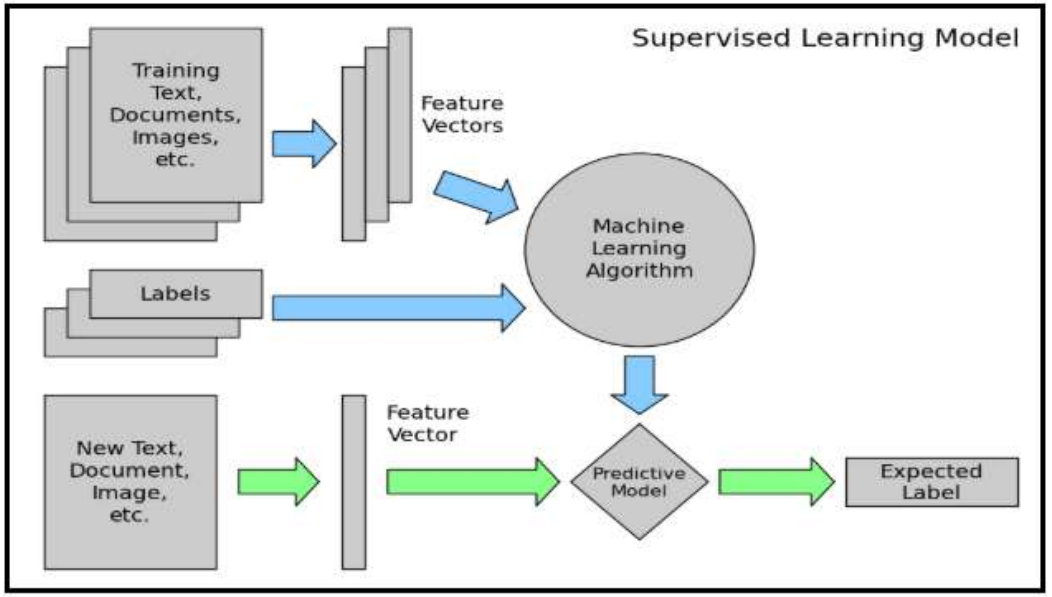
Text Classification

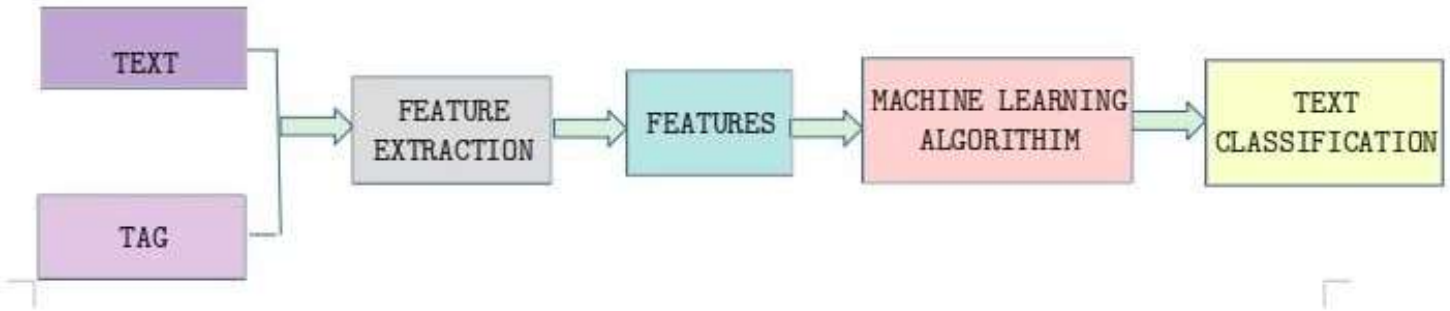
Text Classification

- Text Classification is an example of supervised machine learning task since a labelled dataset containing text documents and their labels is used for train a classifier





- Quite often, we may find ourselves with a set of text data that we'd like to classify according to some parameters (perhaps the subject of each snippet, for example) and text classification is what will help us to do this.
- The diagram below illustrates the big-picture view of what we want to do when classifying text. First, we extract the features we want from our source text (and any tags or metadata it came with), and then we feed our cleaned data into a machine learning algorithm that do the classification for us.



Text and Tag

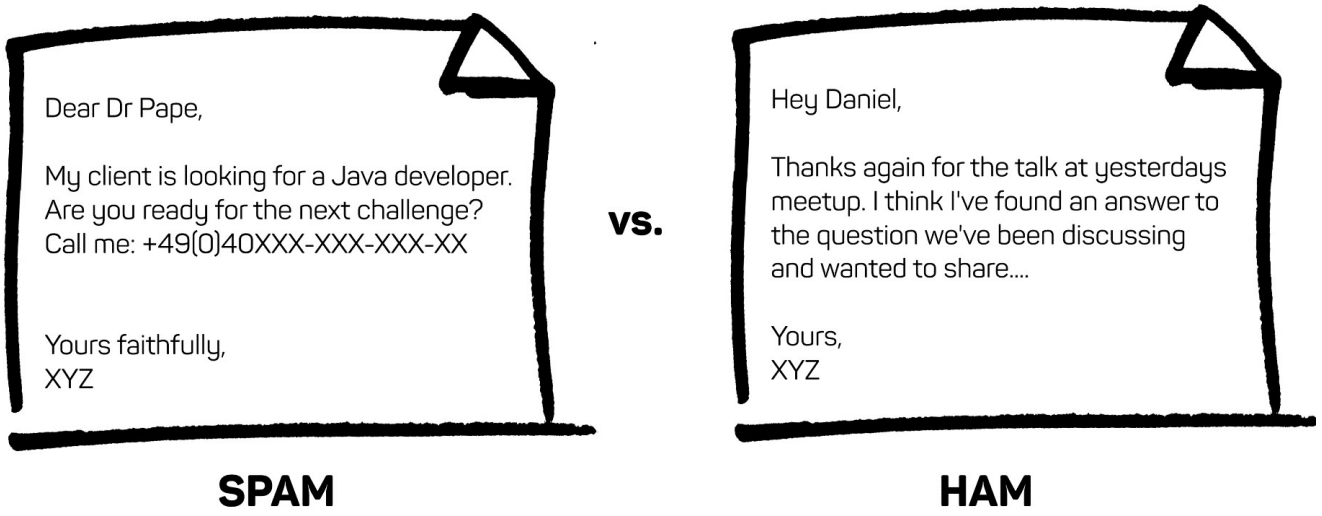
- Sentiment Analysis

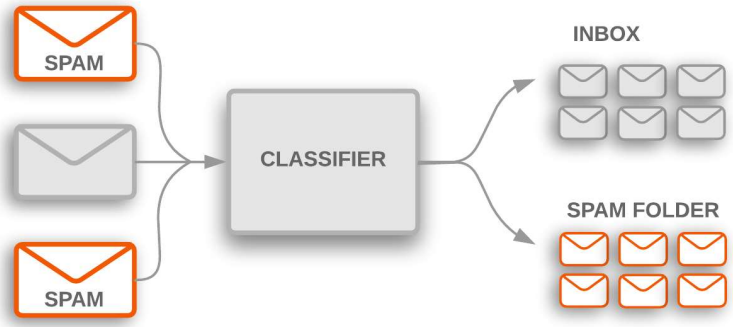
•*'Awful experience. I would never buy this product again!'* → Very Negative
•*'I don't think there is anything I really dislike about the product'* → Neutral
•*'The older interface was much simpler'* → Negative

Opinion	Aspect	Sentiment
"It's "so" easy to use. It took less than a week to understand where everything is in Drift"	UX-UI	Positive
"The mobile app can be really glitchy and is definitely not user friendly"	Mobile App	Negative
"Their customer success team is amazing and there's always someone available from their support team on live chat to help you"	Customer Service	Positive

Text and Tag

- Email classification





Text cleaning/Text preprocessing

- Word Tokenization
- Change all the text to lower case
- Punctuation removal
- Stop words removal
- Lexical normalization – stemming, Lemmatization
- Remove Blank rows in Data, if any
- Remove Non-alpha text

Raw Text	Pre-processed Text
Stuning even for the non-gamer: This sound track was beautiful! It paints the senery in your mind so well I would recomend it even to people who hate video game music! I have played the game Chrono Cross but out of all of the games I have ever played it has the best music! It backs away from crude keyboarding and takes a fresher step with grate guitars and soulful orchestras. It would impress anyone who cares to listen! ^_^	['stun', 'even', 'sound', 'track', 'beautiful', 'paint', 'senery', 'mind', 'well', 'would', 'recomend', 'even', 'people', 'hate', 'video', 'game', 'music', 'play', 'game', 'chrono', 'cross', 'game', 'ever', 'play', 'best', 'music', 'back', 'away', 'crude', 'keyboarding', 'take', 'fresh', 'step', 'grate', 'guitar', 'soulful', 'orchestra', 'would', 'impress', 'anyone', 'care', 'listen']

Train and Test Set

- split up the data into a training set and a testing set.
- The training data set will be used to fit the model and the predictions will be performed on the test data set.

- We can train the model using data which we call as training data or training set. The training data is the one which already has the actual value.
- But how do we know after training the model is overall good ?
  For that, we have test data/test set which is basically a different data for which we know the values but this data was never shown to the model before.
- Thus if the model after training is performing good on test set as well then we can say that the Machine Learning model is good.at the model should have predicted and thus the algorithm changes the value of parameters to account for the data in the training set.

Feature extraction

- **Feature Engineering:** The next step is the Feature Engineering in which the raw dataset is transformed into flat features which can be used in a machine learning model. This step also includes the process of creating new features from the existing data.
- Bag-of-words: Count Vectors as features
- TF-IDF Vectors as features
  - Word level - Matrix representing tf-idf scores of every term in different documents
  - N-Gram level - N-grams are the combination of N terms together. This Matrix representing tf-idf scores of N-grams
  - Character level - Matrix representing tf-idf scores of character level n-grams in the corpus
- Word Embeddings as features
- Text / NLP based features
- Topic Models as features

Document term matrix

- Doc 1: I love dogs.
- Doc 2: I hate dogs and knitting.
- Doc 3: Knitting is my hobby and my passion.
- create a matrix of document and words by counting the occurrence of words in the given document.

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

Tf-IDF

- Term Frequency(TF), you just count the number of words occurred in each document divided by the number of words in that document
- IDF(Inverse Document Frequency) measures the amount of information a given word provides across the document.
- Word with high tf-idf in a document, it is most of the times occurred in given documents and must be absent in the other documents. So the words must be a signature word.

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	0.18	**0.48**	0.18							
Doc 2	0.18		0.18	**0.48**	0.18	0.18				
Doc 3					0.18	0.18	**0.48**	**0.95**	**0.48**	**0.48**

Machine Learning Algorithm/model

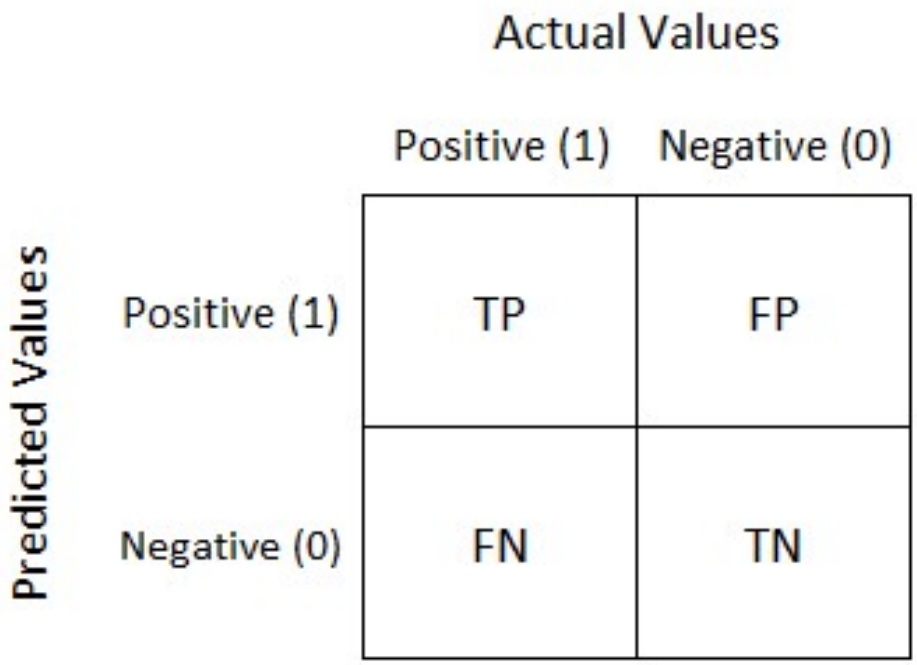
- **Model Training:** The next step is the Model Building step in which a machine learning model is trained on a labelled dataset
  - Naïve bayes classifier
  - Logistic regression
  - Support vector machine
  - Artificial Neural Networks
  - …

Evaluation

- The training dataset trains the model to predict the unknown labels of population data.
- There are multiple algorithms, namely, Logistic regression, K-nearest neighbor, Decision tree, Naive Bayes etc. All these algorithms have their own style of execution and different techniques of prediction.
- But, at the end, we need to find the effectiveness of an algorithm.
- To find the most suitable algorithm for a particular business problem, there are few model evaluation techniques.

Evaluation

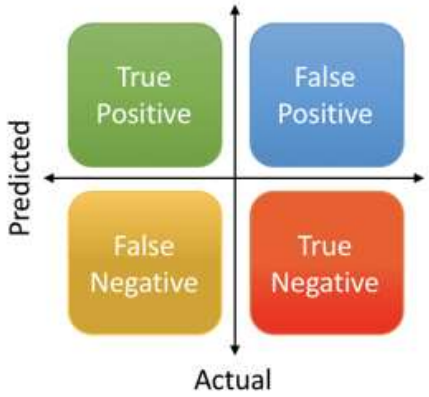
- For a classification task, *positive* means that an instance is labeled as belonging to the class of interest: we may want to automatically gather all news articles about Microsoft out of a news feed, or identify fraudulent credit card transactions, classify emails as spam or not e.t.c.
- A *false positive* is concluding that something is positive when it is not. False positives are sometimes called *Type I errors*.
- A *false negative* is concluding that something is negative when it is not. False negatives are sometimes called *Type II errors*.
- True negative is concluding that something is negative when it is actually negative
- True positive is concluding that something is positive when it is actually positive


Actual Values
Positive (1)
Negative (0)
Predicted Values
Positive (1)
Negative (0)
TP
FP
FN
TN

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$



Evaluation

- precision, recall
  - Precision means the percentage of your results which are relevant.
  - recall refers to the percentage of total relevant results correctly classified by your algorithm
- f-score
  - the harmonic mean of precision and recall:
  - near one when both precision and recall are high, and near zero when they are both low.
  - It is a convenient single score to characterize overall accuracy, especially for comparing the performance of different classifiers.

$$F_1 = \frac{2}{(1/\mathrm{precision} + 1/\mathrm{recall})} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Example

- Classify email as either spam or not spam

Classify email as either spam or not spam

		Actual	
		Spam	**Not spam**
System (Predicted)	spam	12	8
	Not spam	3	77

- True Positive: **12 (You have predicted the positive case correctly!), system predicted spam and the are truly spam.**
- True Negative: **77 (You have predicted negative case correctly!), system predicted not spam and the email are truly not spam.**
- False Positive: **8 (Oh! You have predicted these emails are spam, but in actual they are not spam. This is type-II error in this case.)**
- False Negative: **3 (Oh ho! You have predicted that these three emails are not spam. But actually are spam. This is dangerous! Be careful! This is type-I error in this case.)**

Classify email as either spam or not spam

- Accuracy the ratio of the accurately predicted number and the total number of people which is *(12+77)/100 = 0.89*.
- Precision - the ratio, ***12/(12+8) = 0.6*** is the measure of the accuracy of your model in detecting a person to have the disease.
- Recall - the ratio, 12/(12+3) = 0.8 is the measure of the accuracy of your model to detect a person having disease out of all the people who are having the disease in actual.

		Actual	
		Spam	**Not spam**
System (Predicted)	spam	12	8
	Not spam	3	77