**Naïve Bayes Algorithm**

Strengths

* Easy to implement
* Deals with lot of features set

Weakness

* Can break in case of phrases comprising multiple words

**Support Vector Machine**

Classifier that attempts to maximize the margin of separation from nearest point across multiple classes.

If linear separation is not possible then the hyper-plane can be constructed by transforming the inputs.

Strengths

* Work well with complicated

Weakness

* Training time can be high
* Prone to overfitting in case of noise

SVM C parameter -> lower value of C param avoids overfitting to the test data. Decision boundary becomes more complex when a higher value of C is used i.e. classifier ends up overfitting to the existing data.

**Decision Trees**

**Entropy:** degree of randomness. Alsodegree of impurity in the sample

**Entropy =** Sum[-p \* log p] p = fraction of each input in a class ; log is to base 2

**Information Gain =** entropy(parent) – [weighted average] \* entropy(children)

**St:**

* Easy to use

**We:**

* Prone to over-fitting

**Bias-Variance Trade-off**

**High Bias:** Is not able to learn from data

**High Variance:** Fits perfectly to the test data

**Ensemble Methods:** Meta and many classifiers built from(usually) decision trees

**Adaboost:**

An AdaBoost [1] classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

**Outliers**

**Unsupervised Learnings**

* **Clustering (k-means algorithm)**
  + Hill climbing algorithm and may result in a local minima

**Features Scaling**

* Outliers can mess up as the Xmax & Xmin may be skewed
* Feature scaling = (X-Xmin)/(Xmax-Xmin)
* SVM and K-means clustering algorithms are affected by features scaling

**Text Learning**

* Low information words: also known as stop words
* **NLTK (Natural Language Toolkit)**
* **Text Representation**

1. **Bag of Words**
2. **Stemming**
3. **TfIdf (Term Frequency , Inverse Document Frequency)**
   * Term Frequency(Tf) -> similar to bag of words
   * Inverse Document Frequency (Idf) -> weighted by how often the word occurs in the corpus

**Python Libraries**

1. **numpy ->** numeric function
2. **scipy ->** science function
3. **sklearn ->** machine learning
4. **nltk ->** natural language tool kit
   1. **Stemmer**
5. **re ->** regular expression
6. **pandas ->** dataframe