SPAM MAIL FILTERING

Group No. 36

AAYUSH MAINI
TARUN GUPTA
ADHISH SINGLA

INTRODUCTION

- **Spam**: It is unsolicited and unwanted email from a stranger that is sent in bulk to large mailing lists, usually with some commercial nature sent out in bulk. Some would argue that this definition should be restricted to situations where the receiver is not especially selected to receive the email – this would exclude emails looking for employment or positions as research students for instance. This difficulty in definition demonstrates that the definition depends on the receiver and strengthens the case for personalized spam filtering

- **Ham**: E-mail that is generally desired and isn't considered spam.

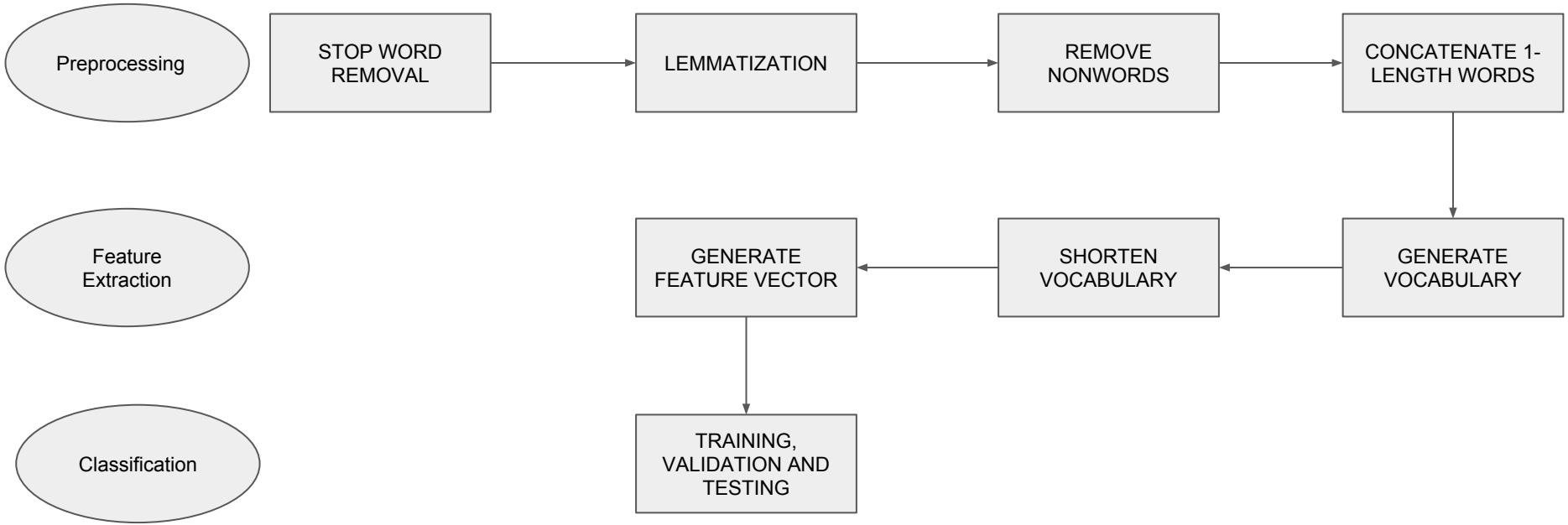
Structure of an E-Mail

- E-Mail mainly consists of **Body** and **Header**.
- **Body** is the message itself.
- The job of the **Header** is to store information about the message and it contains many fields, for example, tracing information about which a message has passed:
  - Received: authors or persons taking responsibility for the message
  - From: intending to show the envelop address of the real sender as opposed to the sender used for replying
  - Return-Path: unique of ID of this message
  - Message-ID: format of content
  - Content-Type: format of content etc.

Spam Filtering

Spam filtering in Internet email can operate at two levels, an individual user level or an enterprise level. An individual user is typically a person working at home and sending and receiving email via an ISP. Such a user who wishes to identify and filter spam email installs a spam filtering system on her individual PC. This system will either interface directly with their existing mail user agent (MUA) (more generally known as the mail reader) or more typically will act as a MUA itself with full functionality for composing and receiving email and for managing mailboxes.

Pipeline


Preprocessing
STOP WORD REMOVAL
LEMMATIZATION
REMOVE NONWORDS
CONCATENATE 1-LENGTH WORDS
Feature Extraction
GENERATE FEATURE VECTOR
SHORTEN VOCABULARY
GENERATE VOCABULARY
Classification
TRAINING, VALIDATION AND TESTING

Pre-Processing (1)

- All words in email have been converted to lowercase.
- Stop Word Removal
  - Words like “and”, “the” and “of” are more commonly used in English sentences.
  - These words are not meaningful in deciding spam/non spam mails.
  - Therefore, these words are removed from the all the mails.
  - We wrote a matlab script to remove such words from the mail.

Pre-Processing (2)

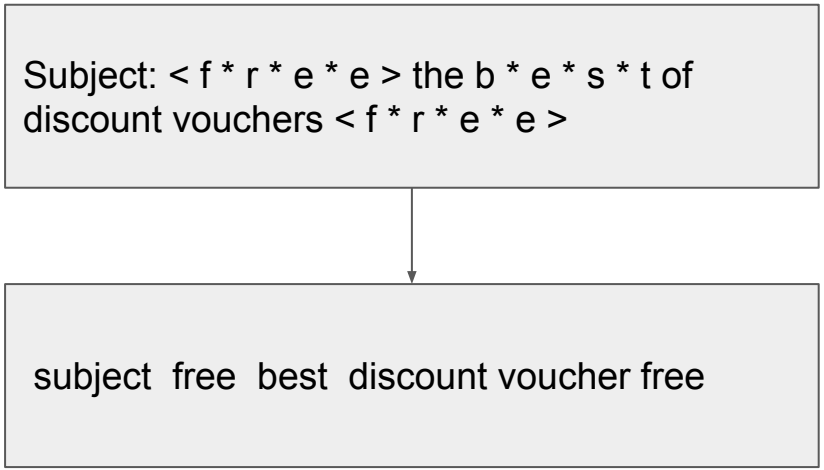
- Lemmatization
  - Words like “include”, “includes” and “included” have the same meaning but different ending.
  - These words will contribute redundant information to feature vector.
  - They have to be adjusted so that they all have the same form.
  - For example, “include”, “includes”, “included” would be represented as “include”.
  - We used direct Matlab implementation of Porter Stemmer Algorithm available here.

Pre-Processing (3)

- Removal of non words
  - Punctuations do not contribute any meaningful and have been removed.
  - All numbers have been reduced to the literal “number”.
  - All white spaces (spaces, tabs and newlines) have been trimmed to a single space character.

Pre-Processing (4)

- Concatenate 1-Length Words
  - All 1-length words are concatenated because in some cases, spammer deliberately introduces spaces to avoid detection.



Pre-Processing Example (5)

- Before Preprocessing

Subject: Re: 5.1344 Native speaker intuitions

The discussion on native speaker intuitions has been extremely interesting, but I worry that my brief intervention may have muddied the waters. I take it that there are a number of separable issues. The first is the extent to which a native speaker is likely to judge a lexical string as grammatical or ungrammatical per se. The second is concerned with the relationships between syntax and interpretation (although even here the distinction may not be entirely clear cut).

- After Preprocessing

re native speaker intuition discussion native speaker intuition extremely interest worry brief intervention muddy waters number separable issue first extent native speaker likely judge lexical string grammatical ungrammatical per se second concern relationship between syntax interpretation although even here distinction entirely clear cut

Feature Extraction (1)

- Generate Vocabulary
  - Vocabulary is the set of all distinct words extracted from preprocessed emails in training data.
  - All tokens are stored in a separate file sorted in lexicographical order.

Feature Extraction (2)

- Reduce Vocabulary Size
  - We need to reduce our vocabulary size
    - To remove very short words (like “aa”, “aaa” etc.) and very long words (which gets generated due to concatenation in Pre-Processing).
    - This will help in removing redundant information from the feature vector and therefore, will reduce the length of feature vector.

Feature Extraction (3)

- Random sampling of words
  - Words from the above vocabulary are randomly sampled to reduce the vocabulary size.
  - To make vocabulary exhaustive, the proportions of words starting with each alphabet (a - z) is kept same, before and after size reduction.
  - For example, if we have 60% words beginning with ‘a’ and 40% words beginning with ‘b’ before size reduction, we will keep the same proportion after size reduction.

Dataset

- LingSpam Dataset
- Divided into 4 types
  - Bare - Non Lemmatized, stop-list disabled
  - Lemm - Lemmatized, stop-list disabled
  - Lemm_stop - Lemmatized, stop-list enabled
  - stop - Non Lemmatized, stop-list enabled
- 2893 training mails
- 5786 test mails

Performance Criterion (1)

- Spam Precision

$$SP = \frac{Number\ of\ Correctly\ Classified\ Spam\ Messages}{Total\ Number\ of\ Messages\ Classified\ as\ Spam}$$

$$= \frac{N_{spam \to spam}}{N_{spam \to spam} + N_{ham \to spam}}$$

Performance Criterion (2)

- Spam Recall

$$SR = \frac{Number\ of\ Correctly\ Classified\ Spam\ Messages}{Total\ Number\ of\ Messages}$$

$$= \frac{N_{spam \to spam}}{N_{spam \to spam} + N_{spam \to ham}}$$

Performance Criterion (3)

- Accuracy

$$A = \frac{Number\ of\ E{-}Mails\ Correctly\ Classified}{Total\ Number\ of\ E{-}Mails}$$

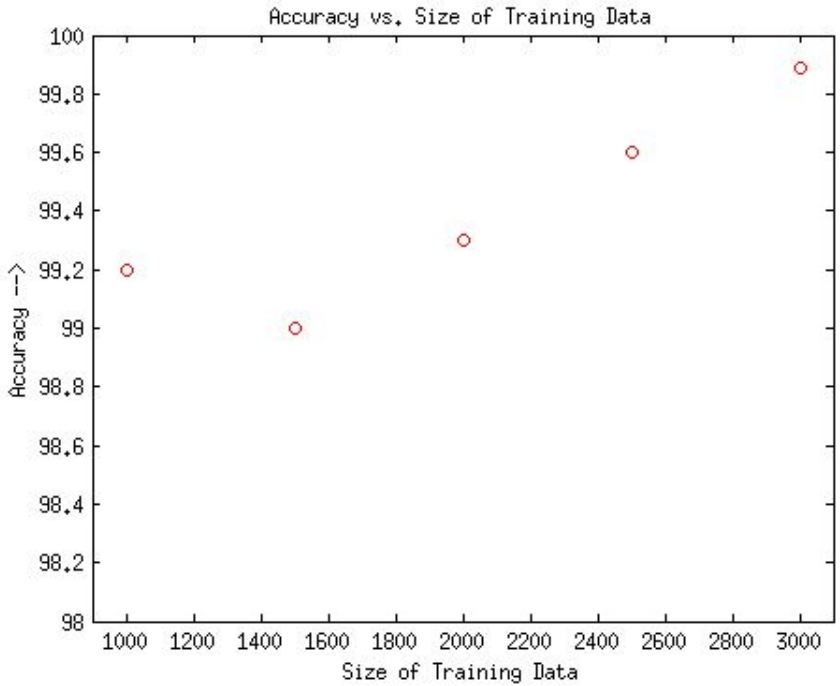
$$= \frac{N_{ham \to ham} + N_{spam \to spam}}{N_{ham} + N_{spam}}$$

Classification

- We will be using following algorithms
  - K - Nearest Neighbours
  - Naive Bayes Classifier
  - Support Vector Machine
  - Artificial Neural Networks
  - Decision Trees
    - CART
    - Random Forests
  - Rough Sets Classifier
  - Adaboosting
  - Online classification
- We have performed 3-Fold cross Validation for each of the algorithms.

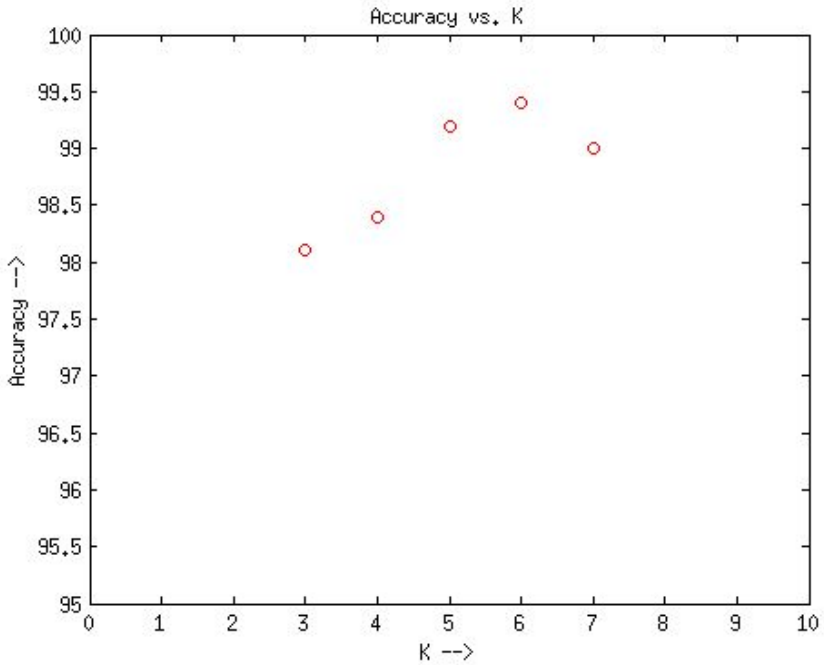
Observation (1)

- Naive Bayes Classifier



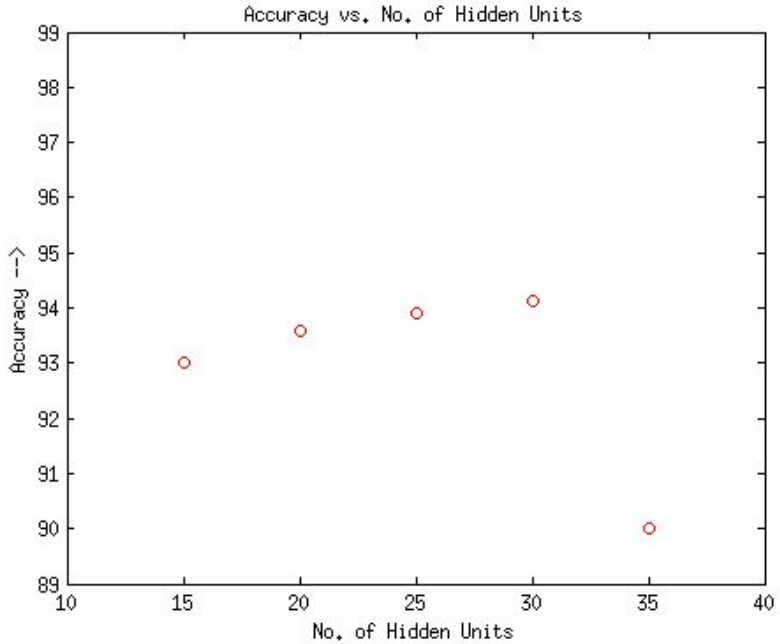
Observation (2)

- K-Nearest Neighbor



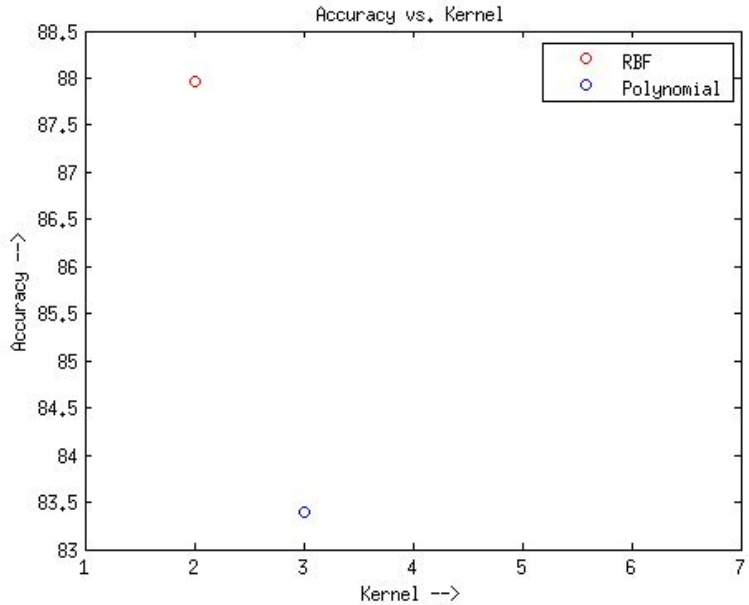
Observation (3)

- Artificial Neural Network



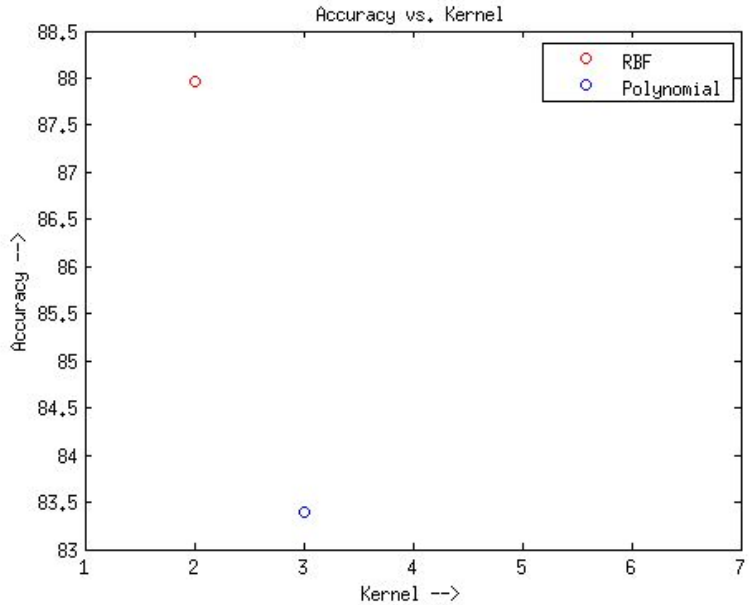
Observation (4)

- Support Vector Machine



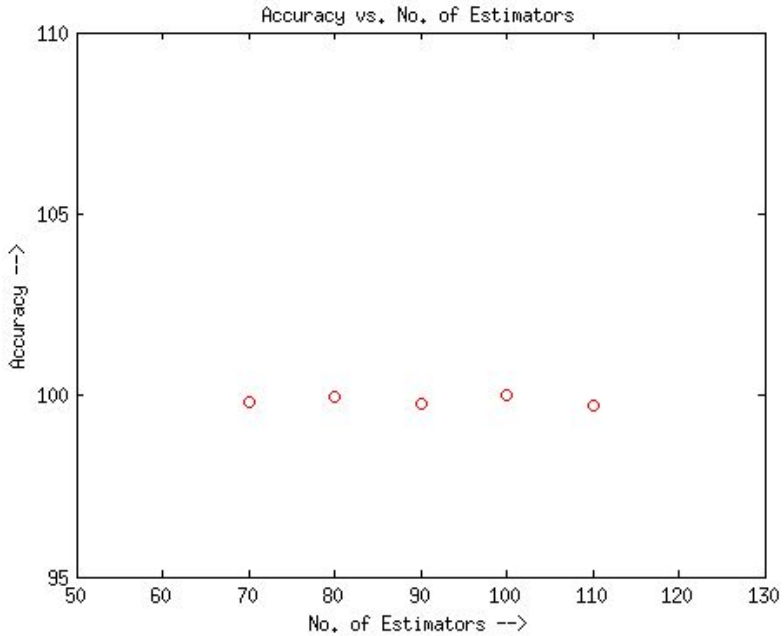
Observation (5)

- Decision Tree



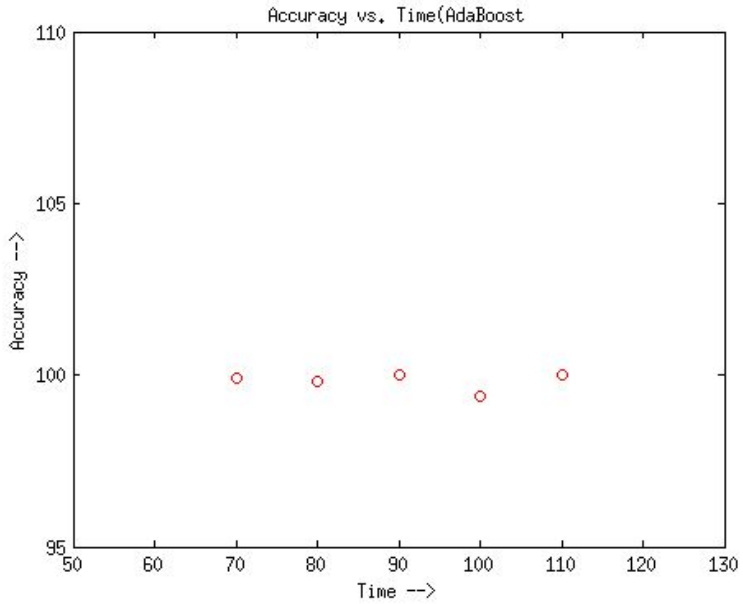
Observation (6)

- AdaBoosting



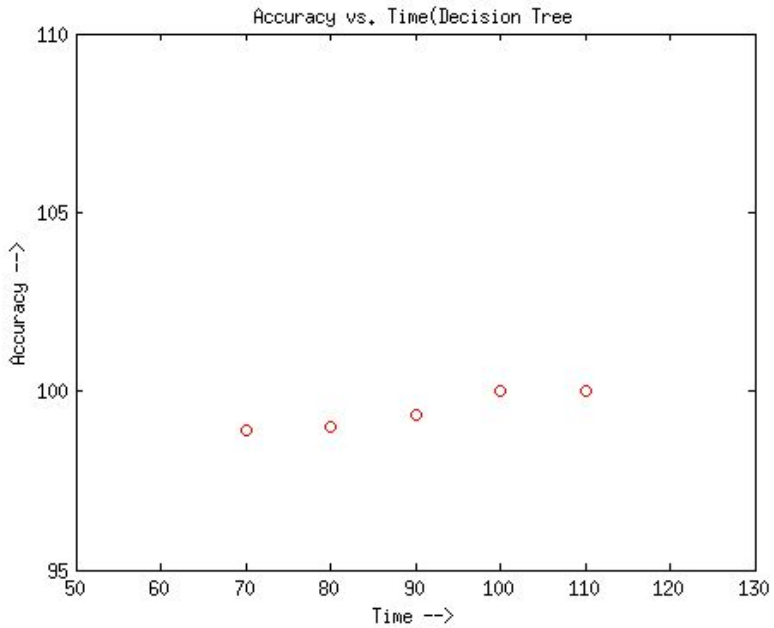
Observation (7)

- Online AdaBoosting



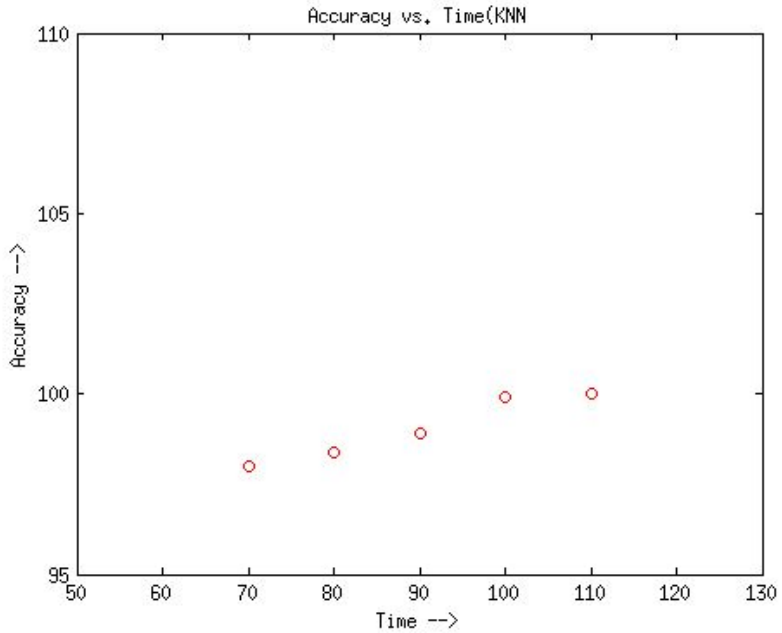
Observation (8)

- Online Decision Tree



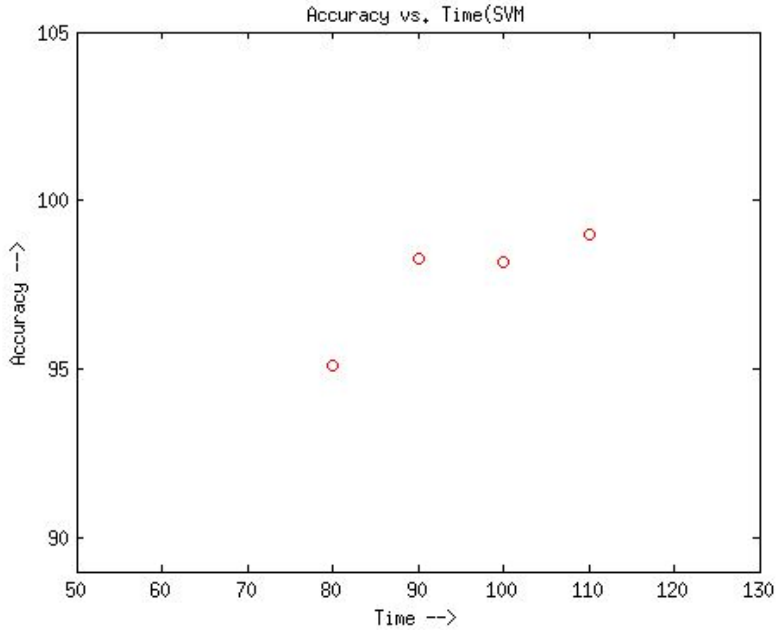
Observation (9)

- Online KNN



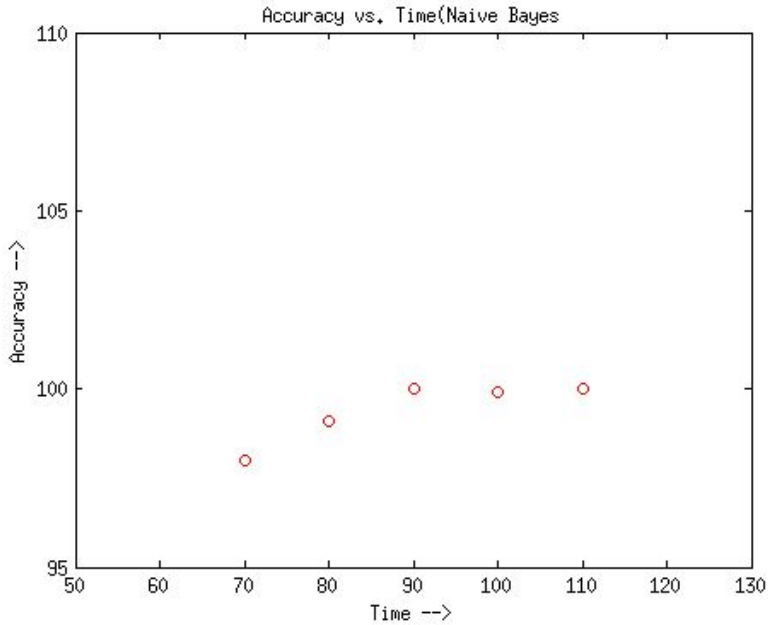
Observation (10)

- Online SVM



Observation (11)

- Online Naive Bayes



Future Scope

- Online Algorithm
  - Fuzzy Kmeans
- Practical Aspects
  - Computational Overload
  - When to cluster again?

Any Questions