# Document Purpose and Audience

* 1. **Purpose**

The goal of this research is to design a multi-label classification model which determines the research topics of a given technical paper. Based on the idea that papers are well organized and some parts of papers are more important than others for text classification, segments such as title, abstract, introduction and conclusion are intensively used in text representation.

This Document explores a simple and efficient baseline for text classification. Our experiments show that our fast text classifier is often on par with deep learning classifiers in terms of accuracy, and many orders of magnitude faster for training and evaluation. We train some algorithms and report some of our figures about the classification.

* 1. **Audience**

This document mainly for machine learning and deep learning developers who interested in text classification and its optimization algorithms. Also researchers in the optimizations of machine learning algorithms on text classification.

# Introduction

Text Classification involves assigning a text document to a set of pre-defined classes automatically, using a machine learning technique. The classification is usually done on the basis of significant words or features extracted from the text document. Since the classes are pre-defined it is a supervised machine learning task. Most of the official communication and documentation maintained in commercial and governmental organizations is in the form of textual electronic documents and e-mails. Much of the personal and other communication done by private individuals is in the form of e-mails, blogs etc. Due to this information overload, efficient classification and retrieval of relevant content has gained significant importance. This paper explains the generic strategy for text classification which includes steps such as pre-processing, feature selection using various statistical or semantic approaches, and modeling using appropriate machine learning techniques (Naïve Bayes, Decision Tree, Neural Network(CNN, RNN), Support Vector Machines, Hybrid techniques). Our paper also discusses some of the major issues involved in automatic text classification such as dealing with unstructured text, handling large number of attributes, examining success of natural language processing based techniques, dealing with missing metadata and choice of a suitable machine learning technique for training a text classifier.

## Software Purpose

The goal of text classification is to automatically classify the text documents into one or more defined categories. Some examples of text classification are:

* Classify text to categories like (Sports, Computer science, Math...etc.).
* Multi-label text classification (title-subtitle…).
* Auto tagging for texts.
* Categorization of articles.

## Software Scope

## This software will be a text classification web app.

* User can input or upload a text and getting the classification results of the document.
* The app should cover different trained models (different techniques).
* The web app will include a research about text classification.

## **Definitions, acronyms, and abbreviations**

|  |  |
| --- | --- |
| Terms & Abbreviations | Definition |
| CNN | Convolutional Neural Network |
| RNN | Recurrent Neural Network |
| NLP | Natural Language processing |
| SVM | Support Vector Machines |

# System Features

## Functional Requirements Users can classify their document by uploading it as a text file or writing it directly. System takes the file and parses its text to sentences, paragraphs and words. System uses a classifier or multi-classifiers to classify the document. System output class or multi-classes of the document.

## **External Interface Requirements**

* 1. User Interfaces
     1. Front-end software: React Native
     2. Back-end software: Django
     3. Database software: MySQL
  2. Hardware Interfaces
     1. Any operating system with web browser
  3. Communications Interfaces
     1. This project supports all types of web browsers.

## **Non Functional Requirements**

|  |  |
| --- | --- |
| **Non-functional**  **Requirement** | **Details** |
| **Performance** | Calculation time and response time should be as little as possible, because one of the software’s features is timesaving.  Any interface between a user and the automated system shall have a maximum response time of 5 seconds unless noted by an exception below (exceptional is optional till now). |
| **Usability** | The system should be easy to use. The user should reach the summarized text with one button press if possible. And one of the software’s features is timesaving.  The system also should be user friendly. |
| **Reliability** | This software will be developed with machine learning, feature engineering and deep learning techniques. So, in this step there is no certain reliable percentage that is measurable. |
| **Scalability** | Analytics platforms must be applicable to a machine or facility of any size. |

# System Models

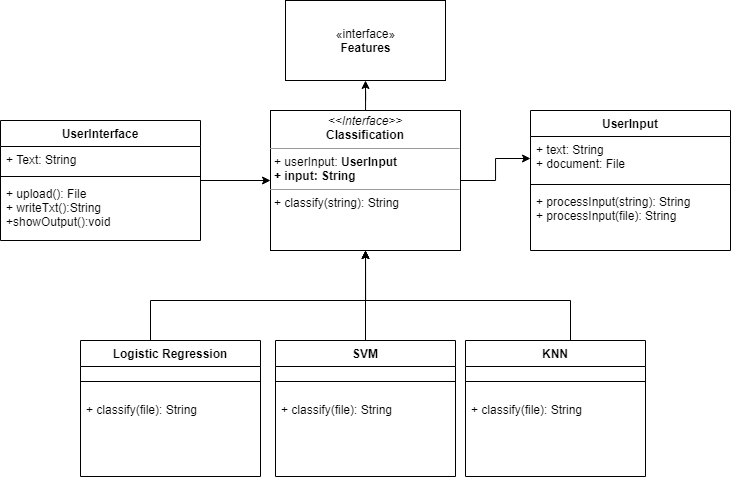
## Use Case Model

## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Upload and classify a Document | |
| Actors: | User | |
| Pre-conditions: | Document Should be text based \*\*\*\*\*\*\* | |
| Post-conditions: | The to be document is ready classified | |
| Flow of events: | **User Action** | **System Action** |
| 1- User uploads a documents or type a text |  |
|  | 2- System verifies user input |
| 3- User request to classify the document or the input text |  |
|  | 4- System classifies the document or the user input. |
| Exceptions: | **User Action** | **System Action** |
| 1- User uploads a non-text input. |  |
|  | 2- Response to user invalid input format. |

## Class Diagram

Login , show history … etc



## Sequence Diagram