

## **Practical 1**

### **Conduct feasibility study for Text Classifier.**

#### **1.1 Feasibility study**

A feasibility study is a study, which is performed by an organization in order to evaluate whether a specific action makes sense from an economic or operational standpoint. The objective of the study is to test the feasibility of a specific action and to determine and define any issues that would argue against this action.

The question a feasibility study essentially tries to answer is: "Should we proceed with the specific action plan?" On top of determining whether the plan is viable/, organizations can use a feasibility study for understanding the risks better and preparing for them.

##### **1.1.1 When should a feasibility study be used**

While feasibility studies are typically conducted by business organizations, other organizations can naturally benefit from it as well. Since the study aims to discover whether an action is viable, it can help organizations to avoid costly or operationally exhausting ventures.

The study is typically used in situations where an important strategic decision needs to be taken.

##### **1.1.2 Core elements of feasibility study**

###### **Technical feasibility**

The first element deals with technical feasibility of the proposed action plan. If your organization is introducing a new product or a service, the technical feasibility study will determine if it's a technically viable action.

Classifier is a service which enables users to get feedbacks of their posts and products. Because of unique nature it has very less competition. You can get feedback reports quite easily without doing much.

Nowadays every business wants to excel that's why this software can be useful for them. Other than that no special requirements are needed

### **Market feasibility**

The second element focuses on testing the market for the proposed action or idea. It examines issues like whether the product or service can be sold at reasonable prices or if there's a marketplace for it. Basically the software is for everyone who want to review their products and posts. But majorly it is targeting users in social media.

There are a lot of similar applications but still no one has completely solved the problem, so this will attract users.

### **Commercial feasibility**

Commercial feasibility is an element of the study focused on the probability of commercial success. It's mainly focused on studying the new business or a new product or service, and whether your organization can create enough profit with it.

The app means helpful to all in many purposes. It can survive without any activity for a short period of time. The cost is usually default values of data, maintaining of application etc. The strength of app is that it's very unique. The privacy of user is not being violated at all unless the user wants to reveal his identity. Although, at higher level, external finance is much needed for its better working.

### **Overall risk assessment**

The fourth element focuses on the major risks the proposed plan can entail. The overall risk assessment part of a feasibility study examines the different ways your organization can reduce the risk of embarking on the new action.

The profits in this software is that with the help of this user can review its contents which means everything to the developers.

### **Operational feasibility**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

If it accurately identifies the posts as negative, positive or neutral for majority of the queries than we can say that it is operationally feasible.

### **Schedule Feasibility**

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Some projects are initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable.

Given our technical expertise, the project deadlines are reasonable and thus have a feasible schedule.

## **Practical 2**

### **Develop software development life cycle (SDLC) for Text Classifier.**

#### **2.1 Introduction**

Software Development Life Cycle (SDLC) is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

SDLC model chosen for the project Text Classifier is Iterative Waterfall Model. Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental). In this incremental model, the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.

The key to a successful use of an iterative software development lifecycle is rigorous validation of requirements, and verification testing of each version of the software against those requirements within each cycle of the model. As the software evolves through successive cycles, tests must be repeated and extended to verify each version of the software.

#### **2.2 Advantages of using Iterative waterfall model**

1. Some working functionality can be developed quickly and early in the life cycle.

2. Results are obtained early and periodically.
3. Parallel development can be planned.
4. Progress can be measured.
5. Risk analysis is better.

## **2.3 Phases**

There are six phases in the iterative waterfall model :

### **2.3.1 Requirement analysis and specification**

Requirements for text classifier project are in following fields:

- Business: In marketing field companies use it to develop their strategies, to understand customer's feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don't buy some products.
- Politics: In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!
- Public Actions: Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

### 2.3.2 Design

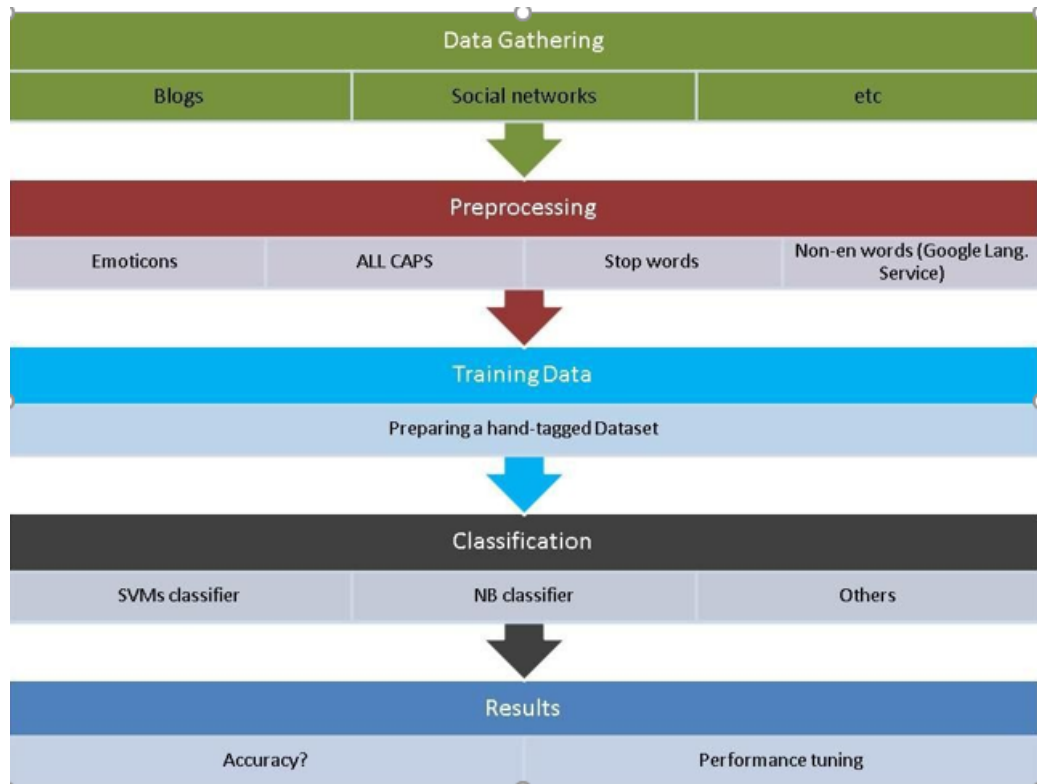


Figure 2.1: Steps involved.

**First Step: Data Collecting :** In this stage data to be analyzed is crawled from various sources like Blogs, Social networks (Twitter, etc.) depending upon the area of application.

**Second Step: Pre-processing :** In this stage, the acquired data is cleaned and made ready for feeding it into the classifier. Cleaning includes extraction of keywords and symbols. For instance Emoticons are the smiley used in textual form to represent emotions e.g. :-), :), =), :D, :-(, :(, =(, ;(, etc.. Correcting the all uppercase and all lowercase to a common case, removing the non-English (or prof-fered language texts), removing un-necessary white spaces and tabs, etc.

**Third Step: Training Data :** A hand-tagged collection of data is prepared by most commonly used crowd-sourcing method. This data is the fuel for the classifier; it will be fed to the algorithm for learning purpose.

**Fourth Step: Classification :** This is the heart of the whole technique. Depending upon the requirement of the application SVM or Naive bayes is deployed for

analysis. The classifier (after completing the training) is ready to be deployed to the real time tweets/text for sentiment extraction purpose.

Fifth Step: Results : Results are plotted based on the type of representation selected i.e. charts, graphs, etc. Performance tuning is done prior to the release of the algorithm.

### **2.3.3 Coding and unit testing**

Coding for this project will be done in python. Following python packages need to be installed additionally:

- Tweepy: tweepy is the python client for the official Twitter API.
- TextBlob: textblob is the python library for processing textual data.

In order to fetch tweets through Twitter API, one needs to register an App through their twitter account and get Consumer Key, Consumer Secret, Access token and Access Token Secret.

We'll test this software on our twitter account feed after it is completely developed.

### **2.3.4 Integration and system testing**

Integration testing (sometimes called integration and testing) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

In text classifier, we will test the same input with various persons manually and their mean value will be compared with the value produced by our software to check its accuracy and correctness.

In system testing, we'll conduct test on a complete, integrated system to evaluate the system's compliance with its specified requirements.

### **2.3.5 Maintenance**

In our text classifier project due to change in requirements and conditions, we will take care of following types of maintenances :

- Corrective Maintenance : This includes modifications and updations done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.

- Adaptive Maintenance : This includes modifications and updations applied to keep the software product up-to date and tuned to the ever changing world of technology and business environment.
- Perfective Maintenance : This includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.
- Preventive Maintenance : This includes modifications and updations to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.



## **Practical 3**

### **Prepare Software Requirement Specification document for Text Analysis**

#### **3.1 Introduction**

##### **3.1.1 Purpose**

The purpose of this document is to outline the requirements which the Text classifier application must meet. The audience of this document includes: project developers and customers, and users who wish to view the project requirements and specifications.

##### **3.1.2 Scope**

This purpose of this project is to develop an automated way of gauging the sentiment of any topic based on social media, specifically Twitter. The software should determine the sentiment of the Twitter community with respect to a given topic, without looking at it manually. The product name will be Text classifier.

##### **3.1.3 References**

- Developer Terms of Service: <https://dev.twitter.com/terms/api-terms>

##### **3.1.4 Overview**

The remaining sections of this document will cover an overall description of this software, as well as requirements and specifications for the software. The overall description of the software is a reference for users, and the detailed specifications and requirements should guide development

## **3.2 System description**

### **3.2.1 Product perspective**

The Text classifier software will take tweets as input. However, only one topic will be analyzed at a time, determining the mood of the Twitter community towards that topic. The application and output will be developed.

The intended user will be a member of the general public who is interested in the sentiment of the Twitter population with respect to various topics.

Twitter does not return any information about the user which that user has not made public. Any personal information that is collected from Twitter will not be stored or used in any way.

### **3.2.2 Assumptions and Dependencies**

An assumption is that it is not possible to accurately determine the sentiment for a 140 character string of English text.

Internet access is required for each analysis session to work properly

## **3.3 Functional requirements**

### **3.3.1 Retrieving Input**

The software will receive three inputs: keywords, analysis session duration, and Tweets.

- Keywords will be entered by the user for each topic.
- The analysis session duration will be set by the user before each session.
- Tweets will be retrieved with the Twitter Streaming API.

### **3.3.2 Real-Time Processing**

The software will take input, process data, and display output in real-time. This will enforce that the snapshot provided by the simple gauge is a current view of the Twitter community's mood on the chosen topic.

The software will generate polarity and classify the text in real time.

### **3.3.3 Sentiment Analysis**

Sentiment analysis will be performed on the user-specified keywords within the Tweet to determine the overall mood of the Tweet relative to the topic. The sen-

timent analysis will provide a negative, neutral, or positive numeric sentiment value.

#### **3.3.4 Output**

The software must output real time data in the form of a simple gauge. In addition, the software may output a graph of polarity trends over time, as well as additional statistics pertaining to a topic (average sentiment over all analysis sessions and total number of tweets processed). This output should be clear and easy to understand.

### **3.4 Non-Functional requirements**

#### **3.4.1 Reliability**

The software will meet all of the functional requirements without any unexpected behavior. At no time should the gauge output display incorrect or outdated information without alerting the user to potential errors.

#### **3.4.2 Availability**

The software will be available at all times on the user's Android device, as long as the device is in proper working order. The functionality of the software will depend on any external services such as internet access that are required. If those services are unavailable, the user should be alerted.

#### **3.4.3 Security**

The software should never disclose any personal information of Twitter users, and should collect no personal information from its own users.

#### **3.4.4 Maintainability**

The software should be written clearly and concisely. The code will be well documented. Particular care will be taken to design the software modularly to ensure that maintenance is easy.

#### **3.4.5 Portability**

This software will be designed to run on any operating system that contains a python compiler.

## **Practical 4**

### **Preparation of software configuration and risk management related documents**

#### **4.1 Software configuration**

The results of a large software development effort typically consist of large number of objects e.g. source code, design document, SRS document, test document, user's manual etc. These objects are usually referred to and modified by a number of software developers throughout the lifecycle of the software. The state of all these objects at any point of time is called the configuration of the software product. The state of each deliverable object changes as development progresses and also as bugs are detected and fixed. Software configuration management deals with effectively tracking and controlling the configuration of a software product during its lifecycle.

##### **4.1.1 Configuration management activities**

A configuration management tool provides automated support for overcoming all the problems. In addition, a configuration management tool helps to keep track of various deliverable objects, so that the project manager can quickly and unambiguously determine the current state of project. The configuration management tool enables the developer to change various components in a controlled manner. Configuration management is carried out through principal activities:

- Configuration identification involves deciding which parts of the system should be kept track of. We can identify the objects into three categories controlled, pre controlled and uncontrolled
- Configuration control ensures that changes to a system happen smoothly. If we make any changes to the input systems then we will get the outputs as per the inputs. If we make changes in one module like user login then same changes must be done smoothly in the user register module.

If we add some feature in our software then it must be updated in user part also.

## **4.2 Risk management**

### **4.2.1 Risk identification**

The project manager needs to anticipate the risks in the project as early as possible so that the impact of the risks can be minimized by making effective risk management plans. A project can be affected by a large variety of risks which can lead down the company badly. There are three main categories of risks which can affect a software project as follows:

- Project risks concern various forms of budgetary, schedule, personnel, resource and customer related problems. In this application it is risk, if the details of one person get revealed to the other person.  
The use of sentiment analysis and related monitoring technology may not be a solution for everyone. "Today it is understood that any information that passes through a corporation in an email, phone conversation or chat session belongs to the corporation and can be used in a regulatory context," said Seth Grimes, an analytics consultant.
- Technical risk concerns potential design, implementation, interfacing, testing and maintenance problems. The risk can be if data does not get entered into the database or wrong entry gets entered into the database table or one table entry gets entered into the other table.  
There are some technical risks involved with sentiment analysis.  
Even in the best of circumstances, it is only 65 to 70 percent accurate, said Susan Etlinger, an analyst with research firm Altimeter Group. She noted that the accuracy rate drops even further when the process is applied to text in languages other than English.
- Business risks : These types of risks include risks of building an excellent product that no one wants, losing budgetary or personnel commitments.  
Companies have to decide whether or not they are comfortable with this level of monitoring done in sentiment analysis. Could there be an adverse reaction from employees if they know they are being monitored? Will it present a challenge to recruitment?

### **4.2.2 Risk Assessment**

The objective of risk assessment is to rank the risks in terms of their damage-causing potential. For risk management, first each risk should be rated in two

ways:

- The likelihood of a risk coming true (r).
- The consequence of the problems associated with that risk (s).

Based on these two factors, the priority of each risk completed:

$$p=r*s$$

where , p is the priority with which the risk must be handled, r is the probability of the risk becoming true, and s is the severity of damage caused due to risk becoming true. If all identified risks are prioritized, then the most likely and damaging risks can be handled first and more comprehensive risk abatement procedures can be designed for these risks.

### **4.2.3 Risk Containment**

After all the identified risks of a project are assessed, plans must be made to first contain the most damaging and the most likely risks. Different risks require different containment procedures. In fact, most risks require ingenuity on the part of the project manager in tackling the risks. There are three main strategies to plan for risk containment:

- **Avoid the risk:** In some cases, you may want to avoid the risk altogether. This could mean not getting involved in a business venture, passing on a project, or skipping a high-risk activity. This is a good option when taking the risk involves no advantage to your organization, or when the cost of addressing the effects is not worthwhile.
- **Share the risk:** You could also opt to share the risk- and the potential gain- with other people, teams, organizations or third parties. For instance, you share the risk when you insure your office building and your inventory with a third party insurance company, or when you partner with another organization in a joint product development initiative.
- **Accept the risk:** Your last option is to accept the risk. This option is usually best when there's nothing you can do to prevent or mitigate a risk, when the potential loss is less than the cost of insuring against the risk, or when the potential gain is worth accepting the

## **Practical 5**

### **Case Study of OpenProj**

#### **5.1 History and current status**

OpenProj was developed at Projity by Marc O'Brien, Howard Katz and Laurent Chretienneau in 2007. It moved out of beta with the release of Version 1.0, on January 10, 2008.

In late 2008, Projity was acquired by Serena Software. As of early 2009 support for OpenProj and communication about development of OpenProj seem to have been suspended. There were actually regressions with a few commits to the CVS, There has been no improvement in the past four years, and it is no longer compatible with Microsoft Project.

In 2012, the founders of OpenProj forked the abandoned code base of OpenProj and started development for a new release. The initial release of this fork occurred in August 2012. The name of the new fork is ProjectLibre

#### **5.2 Features**

The current version includes:

- Earned value costing.
- Gantt chart.
- PERT chart.
- Resource Breakdown Structure (RBS) chart.
- Task usage reports.
- Work Breakdown Structures (WBS) chart.

### **5.3 Popularity**

It has been downloaded over 4,000,000 times in over 142 countries. Three months after the beta version release, on SourceForge an average of 60,000 copies a month were downloaded. With a SourceForge activity percentile of 99.964, at number 15 it was listed just ahead of the popular messaging application Pidgin. In May 2008 the total number of downloads on SourceForge reached 500,000.

### **5.4 Bugs**

As of version 1.4, bugs in the software generally only manifest for users who are attempting more advanced features. For example, tasks may mysteriously start at a certain time (they behave as if they have a 'Start no earlier than' constraint even though none exists, and the project start date is not a constraint), links show gaps, fixed cost for summary tasks neither sums nor is editable, etc. Sometimes these errors are solved by restarting the software, but others are persistent. Compared to Microsoft Project, which it closely emulates, OpenProj has a similar user interface (UI), and a similar approach to construction of a project plan: create an indented task list or work breakdown structure (WBS), set durations, create links (either by (a) mouse drag, (b) selection and then button-down, or (c) manually type in the 'predecessor' column), assign resources. The columns (fields) are the same as for Microsoft Project. Users of either software should be broadly comfortable using the other. Costs are the same: labour, hourly rate, material usage, and fixed costs: these are all provided.

However, there are small differences in the UI (comments apply to version 1.4), which take some adaptation for those familiar with Microsoft Project, i.e. OpenProj can't link upwards with method (c), inserting tasks is more difficult than in Microsoft Project, and OpenProj can't create resources on the fly (have to create them first in the resource sheet). There are also several more serious limitations with OpenProj, the chief of these being the unavailability of more detailed views and reports typical of Microsoft Project. For example, though the fields exist for cost, there is no quick way to show them other than to manually insert them. This requires a relatively advanced user: someone who knows what the fields might be called and how to use them.



## **5.5 Licensing**

Some features of OpenProj are limited to users acquiring a purchased license; for those users using OpenProj for free, a slightly limited feature set is provided. For example, OpenProj(v1.4) does not allow the in-house exporting of PDF output, though the usefulness of such a feature is questionable. It is possible to circumvent the reduced feature set using external software, though as with all paid software, donation or purchase is appreciated by the developers.

## **5.6 Derivatives**

ProjectLibre: The original founders of OpenProj started to develop a complementary server for OpenProj in 2012, comparable to Microsoft Project Server for Microsoft Project. During development they realized, that the fact that OpenProj had not been updated anymore by Serena Software during the last four years will become problematic to their goal, so they needed to develop first a significantly updated version of OpenProj. This version was released as a fork called ProjectLibre in August 2012. The complementary server will be called ProjectLibre Server. ProjectLibre corrects many issues of OpenProj and introduces significant features such as:

- Import/export with Microsoft Project 2010.
- Printing.
- PDF exporting (without any restrictions)
- New ribbon user interface.
- Full compatibility with Microsoft Project 2010.
- Many bug fixes and correction of issues that OpenProj encounters.