

**Title of the Case study**

**Submitted By :**

**Students' Name(Roll no)**

# ABSTRACT

**This should be less than 200 words.**

## ***Informal description (ill posed problem)***

Describe the problem as though you were describing it to a friend or colleague.

***For example: I need a program that will tell me which tweets will get retweets.***

## ***Formal description(Make it well posed problem)***

A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .

## ***For example:***

- ***Task (T):*** Classify a tweet that has not been published as going to get retweets or not.
- ***Experience (E):*** A corpus of tweets for an account where some have retweets and some do not.
- ***Performance (P):*** Classification accuracy, the number of tweets predicted correctly out of all tweets considered as a percentage.

## **Assumptions**

Question everything that comes to your mind and have some assumptions if applicable

# **1. INTRODUCTION**

## **Motivation :**

Consider your motivation for solving the problem. What need will be fulfilled when the problem is solved?

*For example,*

*You may be solving the problem as a learning exercise. This is useful to clarify as you can decide that you don't want to use the most suitable method to solve the problem, but instead you want to explore methods that you are not familiar with in order to learn new skills.*

*Alternatively, you may need to solve the problem as part of a duty at work, ultimately to keep your job.*

## **Benefits of solution :**

Consider the benefits of having the problem solved. What capabilities does it enable?

If it benefits you personally, then be clear on what those benefits are and how you will know when you have got them.

*For example, if it's a tool or utility, then what will you be able to do with that utility that you can't do now and why is that meaningful to you?*

## **Solution Use**

Consider how the solution to the problem will be used and what type of lifetime you expect the solution to have. As programmers we often think the work is done as soon as the program is written, but really the project is just beginning its maintenance lifetime.

The way the solution will be used will influence the nature and requirements of the solution you adopt.

Consider whether you are looking to write a report to present results, or you want to operationalize the solution. If you want to operationalize the solution, consider the functional and nonfunctional requirements you have for a solution, just like a software project.

## **2. Dataset finalization**

Find 3-4 datasets that suit well for the application you chose. Give the source of data also.

Describe each dataset in terms of

1. What is the data about?
2. What are the number of features and describe each of the features and explain the importance?
3. In what all applications the dataset has been used previously, if any.