DAYANANDA SAGAR UNIVERSITY

SOFTWARE EFFORT ESTIMATION USING MACHINE LEARNING TECHNIQUES

A MAJOR PROJECT SYNOPSIS

Submitted By:

B P GAYATHRI ANANYA (ENG17CS0047)

BHARAT NILAM (ENG17CS0050)

CHIRAG P D (ENG17CS0059)

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

at

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING, BANGALORE – 560068

VII Semester

(Course code: 16CS481)

Under the guidance of

Dr. Shyamsundar Pandeya

TABLE OF CONTENTS

1. PROBLEM STATEMENT

Develop an effective effort estimation model achieving best possible accuracy level, optimizing software projects by estimating efforts for the same using machine learning techniques.

2. ABSTRACT

In software engineering, the main aim is to develop a high-quality project that fall within scheduled time and budget, this procedure is called effort estimation. Effort estimation is crucial and important for a company to do because hiring more people than needed will lead to loss of income, and hiring less people than needed will lead to delay of project delivery. The aim of this study is to estimate software effort objectively by using machine learning techniques instead of subjective and time-consuming estimation methods. We would be using decision tree. We are using the boosting algorithm to increase the accuracy level of our ensemble model which is a combination of SVM, decision tree and GLM, ensemble learning will be tried on two public datasets namely Desharnais and Maxwell.

3. INTRODUCTION

Successful project is that the system is delivered on time and within budget and with the required quality.

In software development, effort estimation is the process of predicting the most realistic amount of effort (expressed in terms of person-hours or money) required to develop or maintain software based on incomplete, uncertain and noisy input.

Software researchers and practitioners have been addressing the problems of effort estimation for software development projects since at least the 1960s.

Most of the research has focused on the construction of formal software effort estimation models. The early models were typically based on regression analysis or mathematically derived from theories from other domains. Since then a high number of model building approaches have been evaluated, such as approaches founded on case-based reasoning, classification and regression trees, neural networks, genetic programming etc.

The most common estimation methods today are the parametric estimation models COCOMO, SEER-SEM and SLIM.

The product/software effort/cost-estimation techniques are applied to predict the effort required to finish the project. An incorrect estimation leads to increase in deadline and budget of the project which may further consequence to failure of the project.

Effort estimation is crucial and important for a company to do because hiring more people than needed will lead to loss of income, and hiring less people than needed will lead to delay of project delivery.

The estimation models and techniques are used in different phases of software engineering like budgeting, risk analysis, planning, etc.

4. SOFTWARE AND HARDWARE REQUIREMENTS

a. Functional Requirements

i. Gathering Datasets

- **Desharnais dataset:**
  Desharnais dataset consists of 81 projects collected by J.M. Desharnais in the late 1980s from a Canadian software house. The original dataset consists of 12 attributes: Team experience, Manager experience, year project end, entities, transactions, length, points non adjust, points adjust, adjustment, effort(dependent), project ID and language. Despite the fact that this dataset is now more than 25 years old, it is one of the largest and most used publicly available datasets
- **Maxwell dataset:**
  Maxwell dataset is a relatively new dataset consists of 62 projects between 1985 and 1993. Each project is described by 27 attributes in which all attributes are numerical.
  The attributes are software year, application type, hardware platform, database, user interface, source, telon use, number of languages used, customer participation, Development Environment, Staff Availability Standards Use, Methods Use, Tools Use, Software's Logical Complexity , Requirements Volatility , Quality Requirements, Efficiency Requirements, Installation Requirements, Staff Analysis Skills, Staff Application Knowledge, Staff Tool Skills, Staff Team Skills Duration (months), Application Size (Function Points), Time and effort

ii. Data Processing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues.
The steps involved in data processing are

- Acquiring the dataset and importing all the required libraries for data processing:
  The three main libraries that will be used are:
  - NumPy
  - Pandas and
  - matplotlib
- Identifying and handling the missing values:
  This step involves filling in the missing values with appropriate data like the mean of that attribute to get accurate results.

b. Software Requirements

- Anaconda Environment
- Jupyter Notebook
- Python libraries such as NumPy, pandas, Matplotlib, etc
- Operating system: Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others

c. Hardware Requirements

- Physical server or virtual machine
- System architecture: Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86; Linux- 64-bit x86
- Minimum 5 GB disk space