

A Project Report

On

Predicting of Pulsar Stars using Machine Learning

Submitted in Partial fulfillment of the requirement for the award of the degree of

**BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY**

SUBMITTED

By

Prashanth Yarram

20675A1202

Under the esteemed guidance of

JYOTSNA

PROFESSOR



Department of Information Technology

Accredited by NBA

J.B. Institute of Engineering & Technology

(UGC AUTONOMOUS)

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Baskar Nagar, Yenkapally, Moinabad Mandal, R.R. District, Telangana

(India)-500075 2022-2023

J.B. INSTITUTE OF ENGINEERING & TECHNOLOGY

(UGC AUTONOMOUS)

(Accredited by NAAC, Permanently Affiliated to JNTUH)

Baskar Nagar, Yenkapally, Moinabad Mandal, R.R. Dist. Telangana (India) -500 075

DEPARTMENT OF INFORMATION TECHNOLOGY

Accredited by NBA



CERTIFICATE

This is to certify that the project report entitled “**Predicting of Pulsar Stars using Machine Learning**” being submitted to the Department of Information Technology, J.B. Institute of Engineering and Technology, in accordance with Jawaharlal Nehru Technological University regulations as partial fulfillment required for successful completion of Bachelor of Technology is a record of bonafide work conducted during the academic year 2022-23 by,

PRASHANTH YARRAM - 20675A1202

Internal Guide

J.Jyotsna

ASSISTANT PROFESSOR

Head of the Department

Dr. L. Sridhara Rao

ASSOCIATE PROFESSOR

J.B. INSTITUTE OF ENGINEERING & TECHNOLOGY

(UGC AUTONOMOUS)

(Accredited by NAAC, Permanently Affiliated to JNTUH)

Baskar Nagar, Yenkapally, Moinabad Mandal, R.R. Dist. Telangana (India) -500 075

DEPARTMENT OF INFORMATION TECHNOLOGY

Accredited by NBA



DECLARATION

We hereby certify that the Main Project report entitled **“Predicting of Pulsar Star using Machine Learning”** conducted under the guidance of **J. JYOTSNA, Professor** is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Information Technology**. This is a record of bonafide work carried out by us and the results embodied in this project report have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Date:

Place:

PRASHANTH YARRAM - 20675A1202

ACKNOWLEDGEMENT

At outset we express our gratitude to almighty lord for showering his grace and blessings upon us to complete this Main Project. Although our name appears on the cover of this book, many people had contributed in some form or the other to this project Development. We could not have done this Project without the assistance or support of each of the following.

First of all we are highly indebted to **Dr. P. C. KRISHNAMACHARY**, Principal for giving us the permission to carry out this Main Project.

We would like to thank **Dr. L. Sridhara Rao**, Associate Professor & Head of the Department of INFORMATION TECHNOLOGY, for being moral support throughout the period of the study in the Department.

We are grateful to **J. Jyotsna**, Professor of the Department of INFORMATION TECHNOLOGY, for his valuable suggestions and guidance given by him during the execution of this Project work.

We would like to thank Teaching and Non-Teaching Staff of Department of Information Technology for sharing their knowledge with us.

PRASHANTH YARRAM - 20675A1202

ABSTRACT

A pulsar is an exceptionally polarized pivoting neutron star or white smaller person that emits a light emission radiation. This radiation can be watched just when the light emission is highlighting Earth and is answerable for the beat appearance of emanation. To predict these pulsar stars, we are implementing the model using some machine learning algorithms like decision tree, random forest, etc. Pulsars are one of the possibilities for the wellspring of ultra-high-vitality enormous beams. The recent exponential growth in the data volume and number of identified pulsar stars is due to pulsar candidate search experiments and surveys. In this study, we investigated the existing methods and techniques used for pulsar prediction, such as applying filters based on pulsar observations, which can adversely affect the success of accurate pulsar prediction. Some of the existing methods are not capable of dealing with large volumes of data and others fail to accurately select the best candidates from pulsar observations. Thus, we developed a new approach based on the traditional supervised machine learning algorithm, which yields faster and more accurate results. We will try various models, including random forest classifier, support vector machine to predict pulsar stars, to get high accuracy, precision, and recall from all those models.

CONTENTS

Sl. No	Name of the Topic	Page No
	Certificate	I
	Declaration	II
	Acknowledgement	III
	Abstract	IV
1.	INTRODUCTION	1
2.	LITERATURE SURVEY	2
3.	SYSTEM ANALYSIS	
	3.1 Existing System	4
	3.2 Drawbacks of Existing System	4
	3.3 Proposed System	4
	3.4 Advantages of Proposed System	4
	3.5 Software Development Life Cycle Model (SDLC)	5
	3.6 Project Implementation Plan	8
4.	SOFTWARE REQUIREMENT SPECIFICATIONS	9
	4.1 Functional Requirements	9
	4.2 Non-Functional Requirements	9
	4.3 Software Requirement Specifications	10
	4.4 Hardware Requirement Specifications	
5.	SYSTEM DESIGN	
	5.1 System Architecture.	11
	5.2 Design Tool Used	12
	5.3 UML Diagrams	15
	5.3.1 Use Case Diagram	15
	5.3.2 Class Diagram	16
	5.3.3 Deployment Diagram	17
	5.3.4 Sequence Diagram	18
	5.3.5 Component Diagram	19
6.	IMPLEMENTATION	
	6.1 Introduction	20
	6.2 Technology Used	21
	6.3 Coding Standards	22

7.	SYSTEM TESTING	
	7.1 Introduction	25
	7.2 Software Testing	25
	7.3 Test Cases	28
	7.4 Bug Report	29
8.	RESULT SCREENS	30
9.	CONCLUSION AND FUTURE SCOPE	44
10.	BIBLIOGRAPHIES	
	References	45
11.	APPENDIXES	
	11.1 Sample Code	47

LIST OF FIGURES

Sl. No	Desperation of Figure	Page No
1	Spiral Model	7
2	Duration graph	8
3	System Architecture	11
4	Visual Paradigm Running on Windows 11	13
5	Use Case Diagram	14
6	Class Diagram	15
7	Sequence Diagram	16
8	Deployment Diagram	17
9	Component Diagram	18
10	DMAIC Flow Chart	22
11	Data set	28
12	Data Analysis	29
13	Attribute Information Graph	30
14	Portion of target variable in data set	31
15	Distribution of values of each of the features	32
17	Correlation Matrix	33
18	Decision tree	34
19	Random Forest Classifier	35
20	Support vector Machine	36
21	Input Screen	37
22	Result Screen	38

LIST OF TABLES

Sl. No	Desperation of Tables	Page no.
1	Test cases	27
2	Bug Report	27

1. INTRODUCTION

1.1 MOTIVATION

Pulsars are not common and automating the process of identifying them would help astronomers and physicist study them: they have also been used to study nuclear physics, General Relativity, and they even helped prove the existence of gravitational waves. In this paper, we are implementing four models to consequently distinguish the presence of pulsar stars. The list of capabilities contains mean, standard deviation, overabundance kurtosis, and skewness of the incorporated profile and mean, standard deviation, abundance kurtosis, and skewness of the DMSNR bend.

A pulsar is a quickly pivoting neutron star. A neutron star is one of the endpoints of the life of a massive star after it explodes in a supernova explosion. A neutron star is one of the end purposes of the life of a gigantic star after it detonates in a supernova blast. This magnetic field is not aligned with the rotation axis of the neutron star. To watch these beats of radiation at whatever point the attractive post is obvious. The beats come at a similar rate as the revolution of the neutron star, and, along these lines, seem occasional. In spite of, before long, radio repeat impedance and fuss can create signals that resemble that of pulsars, so it's particularly intriguing to think of an approach to characterize among pulsars and radio recurrence obstruction or noise and Machine Learning Technology proves to be a perfect stop to simplify and automate this manual task.

1.2 OBJECTIVE OF THE PROJECT

The objective of this project is to develop a supervised machine learning model that can predict a pulsar star. We are utilizing four machine learning models to automatically detect the existence of pulsar stars. The feature set contains mean, standard deviation, excess kurtosis, and skewness of the integrated profile and mean, standard deviation, excess kurtosis, and skewness of the DMSNR curve. Machine learning tools are now being used to automatically label pulsar candidates to facilitate rapid analysis. Classification systems in particular are being widely adopted, which treat the candidate data sets as binary classification problems.

2. LITERATURE SURVEY

In the past, a number of authors and experts have made predictions about pulsars in space. There have been numerous investigations, including those using machine learning techniques, to find these pulsars.

In his paper, N. Obody [2] discussed how well support vector machines (SVM) and artificial neural networks (ANN) performed in predicting pulsars from the same dataset as this paper. He gave a thorough analysis of the two approaches. He came to the conclusion that both SVM neural networks with linear kernels had a 98% accuracy rate.

Nevertheless, he came to the conclusion that none of the approaches stood out as a superior choice.

In their article, Zhen Hong Shang et al.[3] provided three classification algorithms and discussed how well they performed when it came to pulsar detection. Techniques for categorization based on decision trees, support vector machines, and neural networks were all presented

In their article, P. Mounika et al. [4] examined the efficacy of four machine learning models for pulsar classification.

They discussed k-nearest neighbour, support vector machines, random forests, and decision tree classifier (KNN).

They came to the conclusion that the KNN performed better than the other models.

The EPN pulsar dataset was used by Amitesh Singh et al. [5] in their research, and machine learning regression algorithms like decision tree regressor, k-nearest neighbour regressor, and support vector

Pulsar prediction using the vector regressor and other methods. To discover the most effective algorithm for the job, they compared the regressors' performances. In order to cut down on the number of periods, they adopted FFA procedures.

For the aforementioned job, a variety of alternative machine learning techniques can be applied. In order to analyse their performance and determine which algorithm will work best for classifying pulsars, this project integrates all machine learning classification algorithms under one roof.

FEASIBILITY STUDY

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that are analyzed are

Operational Feasibility

Economic Feasibility

Technical Feasibility

Operational Feasibility

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

Economic Feasibility

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer-based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at any time. The Virtual Private Network is to be developed using the existing resources of the organization. So, the project is economically feasible.

Technical Feasibility

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been conducted. The system is technically feasible for development and can be developed with the existing facility.

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

One of the challenges experienced during the scans for pulsars in the previous fifty years is the investigation required for the expanding number of 'competitor' pulsar locations emerging from an expanding volume of information to be looked at. From basic channels to another principled continuous characterization approach utilize factual grouping to separate 8 new highlights to limit the number of highlights without reducing arrangement execution. Every up-and-comer is a potential recognition of a pulsar signal, which displays explicit qualities of intrigue. Precise choices must be made. Mix-ups can prompt the expensive misuse of telescope time on fake signs, and more regrettable, missing legitimate signs altogether.

3.2 Drawbacks of Existing System

- It an Expensive process to find pulsar star using telescopes as there are many fake signs which waste our precious research time.
- Searching for pulsars is a very labor-intensive process, currently requiring skilled people to examine and interpret plots of data output by analysis programs.

3.3 PROPOSED SYSTEM

In the proposed system, we will be using the different parameters to detect the pulsar star that are already calculated and given in dataset and using this dataset we will training multiple machine learning algorithms and evaluate them to get a best model to detect pulsar star.

3.4 Advantages of Proposed System

- Reducing the human labor by using the intelligence and computational power of the machine.
- It is a very cost-efficient method as it does not require a lot of to detect a pulsar star.

- In the proposed machine learning model for detecting pulsar candidates, it is proved that the system is both faster and accurate and dependable.
- Since it reduces the necessity of large number of skilled labors, it is cost efficient.
- Henceforth, this model could be a possible and a better replacement of the procedure that has been followed through years to detect pulsars.

3.5 Software Development Life Cycle Model

SDLC METHDOLOGIES

This document plays a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
 - Evaluating the first prototype in terms of its strengths, weakness, and risks.

- Defining the requirements of the second prototype.
 - Planning a designing the second prototype.
 - Constructing and evaluating the second prototype.
- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
 - The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
 - The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
 - The final system is constructed, based on the refined prototype.
 - The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

The following diagram shows how a spiral model act like:

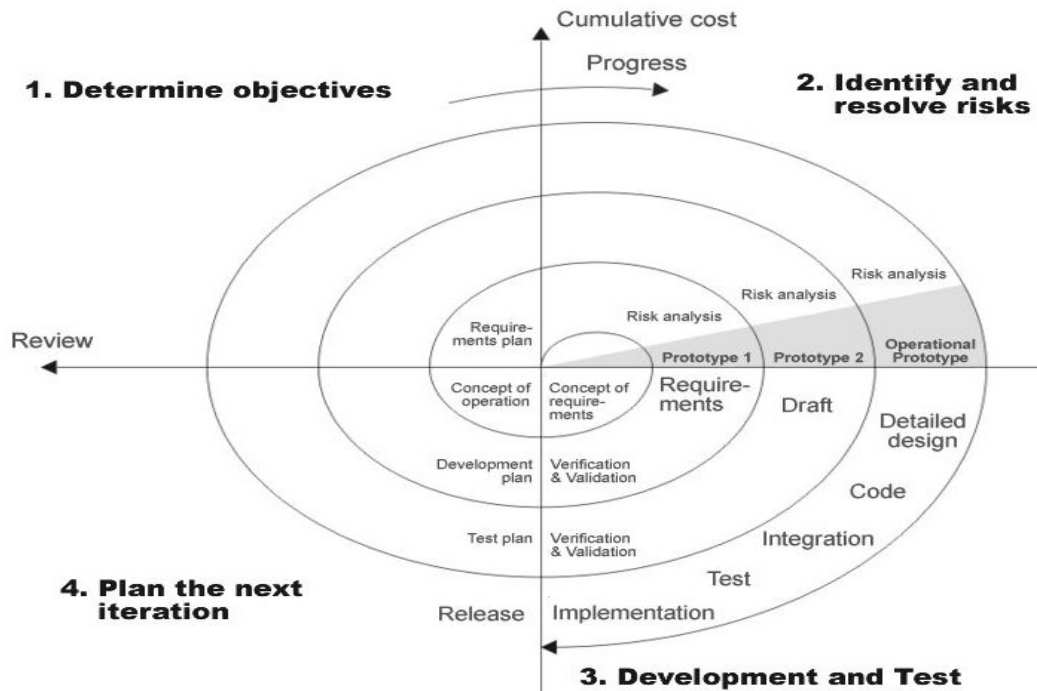


Fig: 3.5.1 spiral model

Advantages

- Estimates(i.e. budget, schedule etc .) become more realistic as work progresses, because important issues are discovered earlier.
- It is more able to cope with the changes that are software development generally entails.
- Software engineers can get their hands in and start working on the core of a project earlier.

3.6 Project Implementation Plan

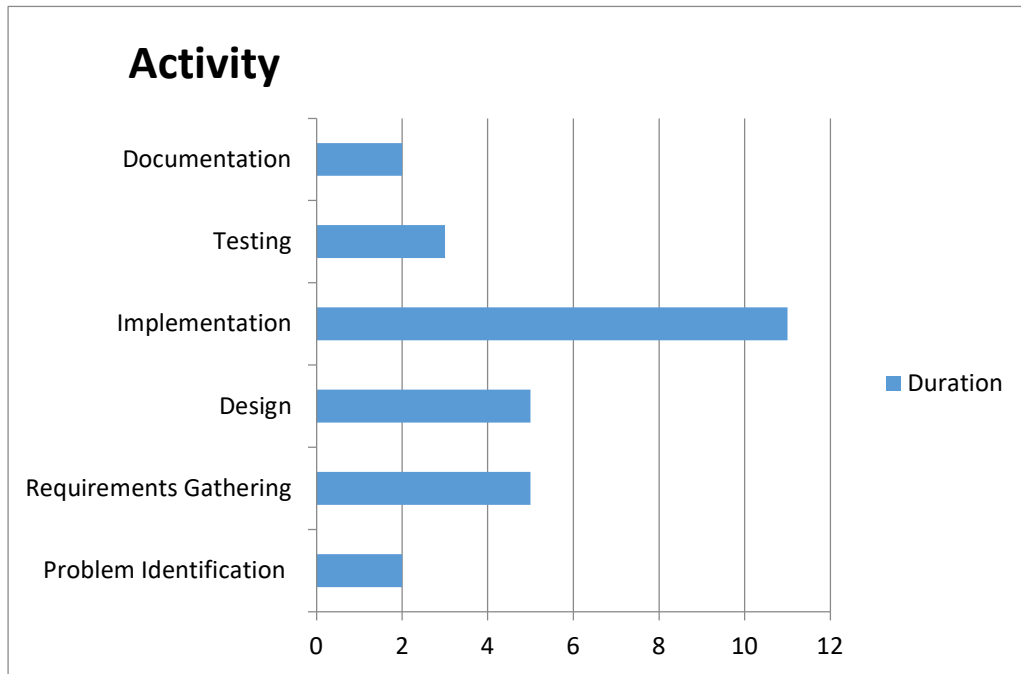


Fig: 3.6.1 Duration graph

4. SOFTWARE REQUIREMENT SPECIFICATIONS

4.1 Functional Requirements

- Ability to process the data to find the data of the star that resembles the pulsar star.
- Ability to differ a star which is not a pulsar star from one that is a pulsar star.
- Able to visualize the results to end users.

4.1 Non-Functional Requirements

- **Accurate results**

Accuracy score in machine learning is an evaluation metric that measures the number of correct predictions made by a model in relation to the total number of predictions made. We calculate it by dividing the number of correct predictions by the total number of predictions.

- **Compatibility**

A compatibility test is an assessment used to ensure a software application is properly working across different browsers, databases, operating systems (OS), mobile devices, networks and hardware.

- **Fast Response**

The system must be able to deliver the response to the user in very less time making the user interaction fast and interactive.

4.3 Software Requirements

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks, tracking the teams, and tracking the team's progress throughout the development activity.

- Language: Python 3.5.
- IDE: Anaconda Navigator, VS Code.
- OS: Windows 7 or above.
- Packages: NumPy, Pandas, matplotlib, seaborn, sci-kit learn

4.4 Hardware Requirement

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the entire system. They are used by software engineers as the starting point for the system design. It should what the system does and not how it should be implemented.

- Processor : I 5
- Hard Disk : 512GB
- Ram : 8GB
- CPU :2GHz
- Processor : Intel i3 or Higher
- Architecture : 32bits or 64bits

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE:

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

Visual Paradigm Online Free Edition

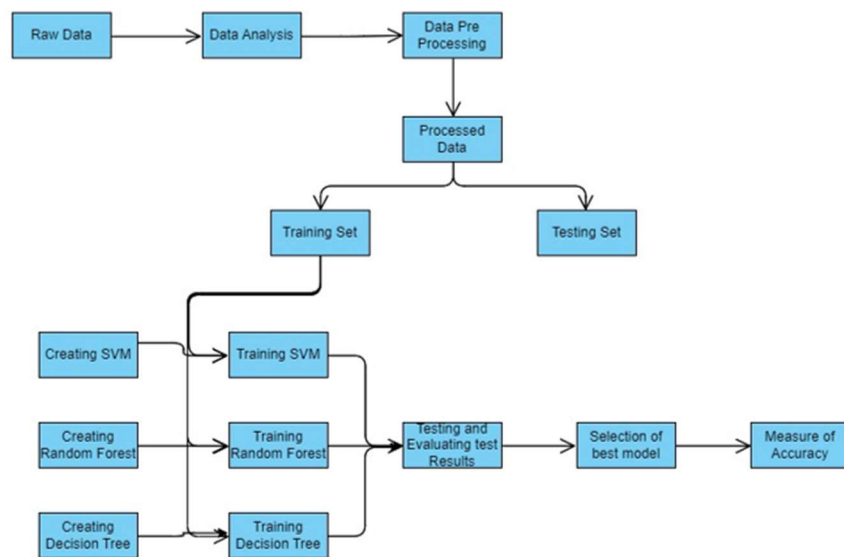


Fig5.1.1 System Architecture diagram

Data Analysis: Data Analysis means understanding the basics of the data being loaded. To have knowledge of number of row and columns, type of data each column has, their statistics and Graphical Structures. So that we can perform Data pre-processing step easily.

Data pre-processing: Data pre-processing means cleaning and preparing the data for giving it as input to the algorithm

1. Cleaning: removing or handling the empty values in the dataset.
2. Dimensionality reduction: reducing the number of Features or Columns in dataset to reduce the computational burden on the hardware and remove columns and features that are useless in solving the classification problem.

Data Splitting: Data Splitting means creating the training set and testing set so that we can train the algorithm and understand the performance of that model. Creating and training algorithms

Creating: instantiating the multiple algorithms which can accept input and produce output and supplying them with the train data to start the training.

Training: Making the particular algorithm understand the training data and become intelligent in that concept.

Testing: Process used to predict the outputs for the inputs in the test set. And understand the performance of Trained Model.

5.2 Design Tool Used - Visual Paradigm Tool

Introduction

Visual Paradigm (VP-UML) is a UML CASE Tool supporting UML2, SysML and Business Process Modeling Notation (BPMN) from the Object Management Group (OMG). In addition, modeling support, it provides report generation and code engineering capabilities including code generation. It can reverse engineer diagrams from code, and provide round-trip engineering for various programming languages.

Contents

- Product Editions
- UML Modeling
- Requirements Management
- Business Process Modeling
- Data Modeling

Product Editions

Higher-priced editions provide more features.

The following editions were available:

- Community Edition
- Modeler Edition
- Standard Edition
- Professional Edition
- Enterprise Edition

UML Modeling

Visual Paradigm supports 13 types of diagrams:

- Class Diagram 13
- Use Case Diagram
- Sequence Diagram
- Communication Diagram
- State Machine Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram
- Package Diagram
- Object Diagram
- Composite Structure Diagram
- Profile Diagram
- Timing Diagram
- Interaction Overview Diagram

Requirements Management

- Visual Paradigm supports requirements management including user stories, use cases, SysML requirement diagrams and textual analysis.
- A SysML requirement diagram specifies the capability or condition that must be delivered in the target system. Capability refers to the functions that the system must support. Condition means that the system should be able to run or produce the result given a specific constraint. Visual Paradigm provides a SysML requirement diagram for specifying and analyzing requirements

Business Modeling

Supports BPMN 2.0 for modeling of business processes. The latest version (Aug 2016) also supports Case Management with CMMN

Data Modeling

Visual Paradigm supports both Entity Relationship Diagrams (ERD) and Object Relational Mapping Diagrams (ORMD). ERD is used to model the relational database. ORMD is one of the tools to show the mapping between class from object-oriented world and entity in relational database world.

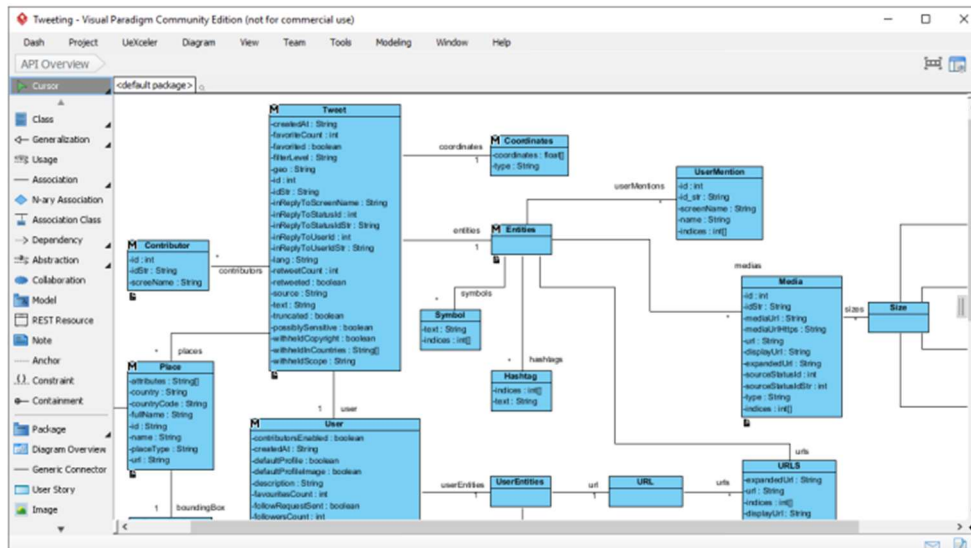


Fig 5.2: Visual paradigm

Visual Paradigm 17.0 running on Windows 11

Developer(s) Visual Paradigm International Ltd.

Initial Release 20 June 2002; 15 years ago 15

Stable Release 17.0 / August 01, 2022

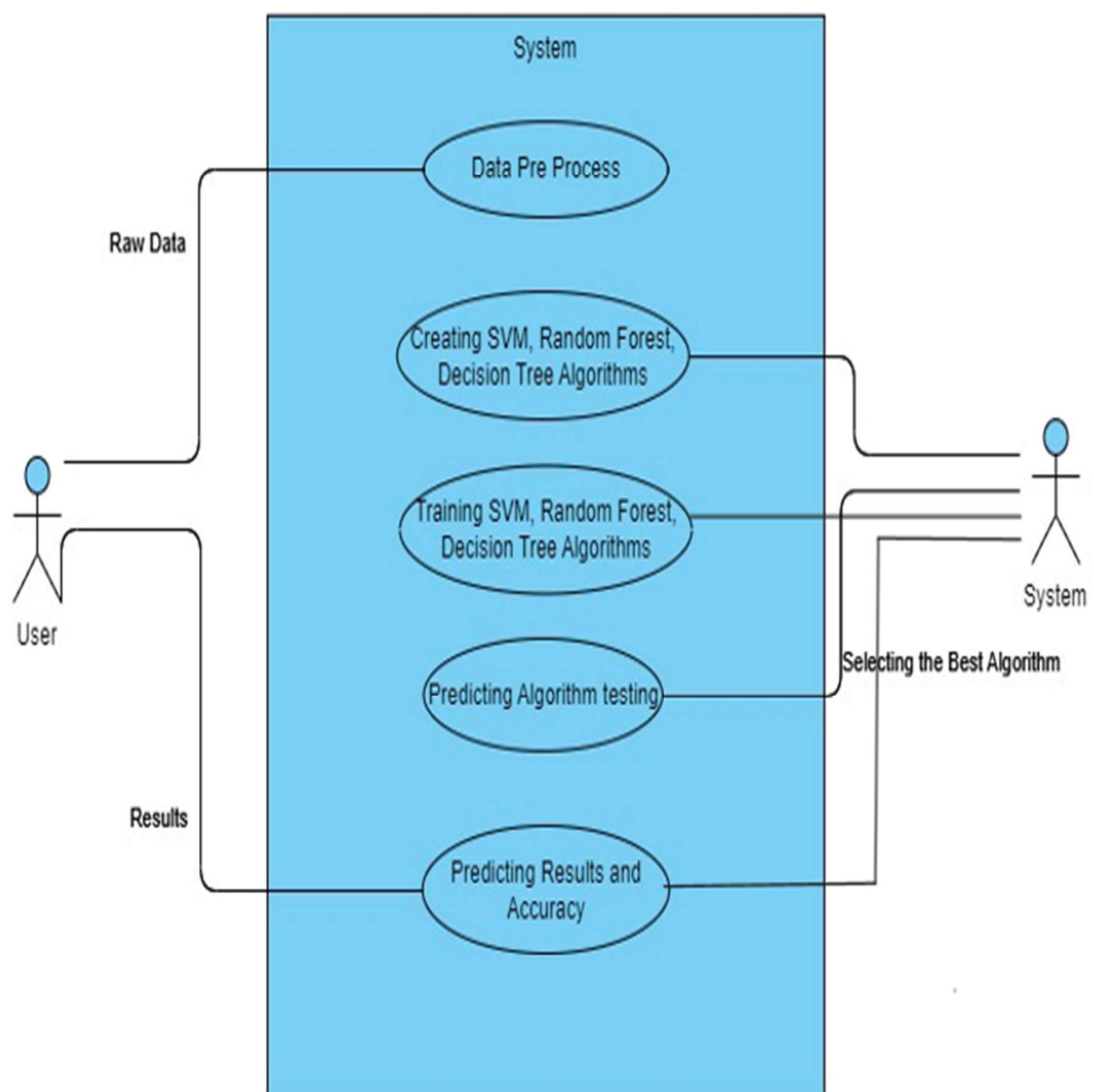
License Proprietary with Free Community Edition

5.3 UML DIAGRAMS

5.3.1 USE CASE DIAGRAM

It shows the set of use cases, actors & their relationships. In our project we have 4 actors sender and receiver & use cases shows encryption and decryption, login, generating key process

Visual Paradigm Online Free Edition



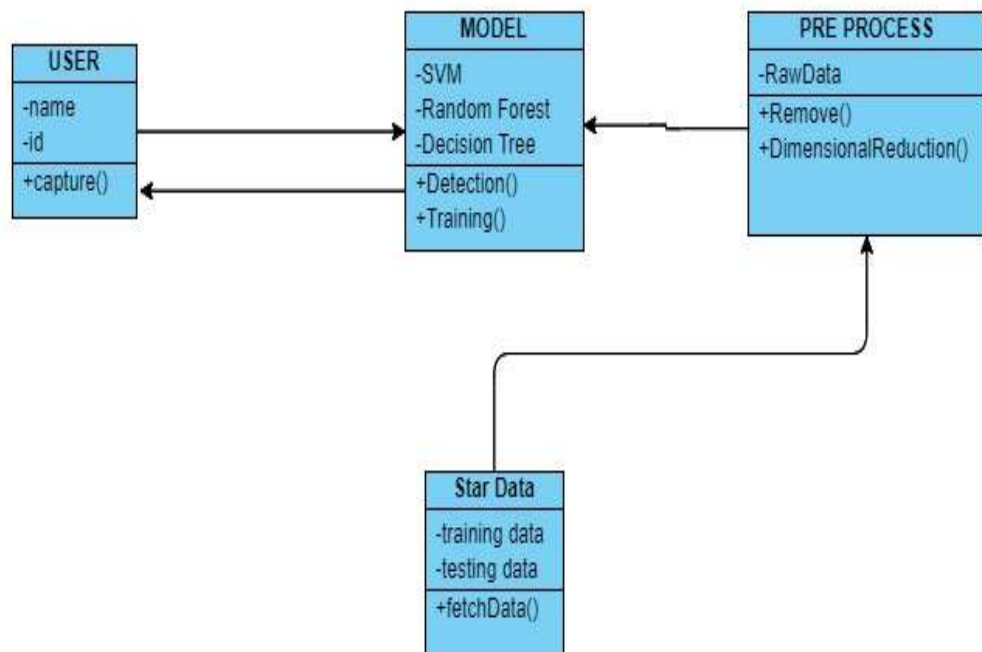
Visual Paradigm Online Free Edition

Fig5.3.1 Use case diagram

5.3.2 CLASS DIAGRAM

A class diagram shows a set of classes, interfaces, and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagrams address the static design view of a system. Class diagrams that include active classes address the static process view of a system.

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Fig 5.3.2 class diagram

5.3.3 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the hardware processors/ nodes/ devices of a system, the links of communication between them and the placement of software files on that hardware.

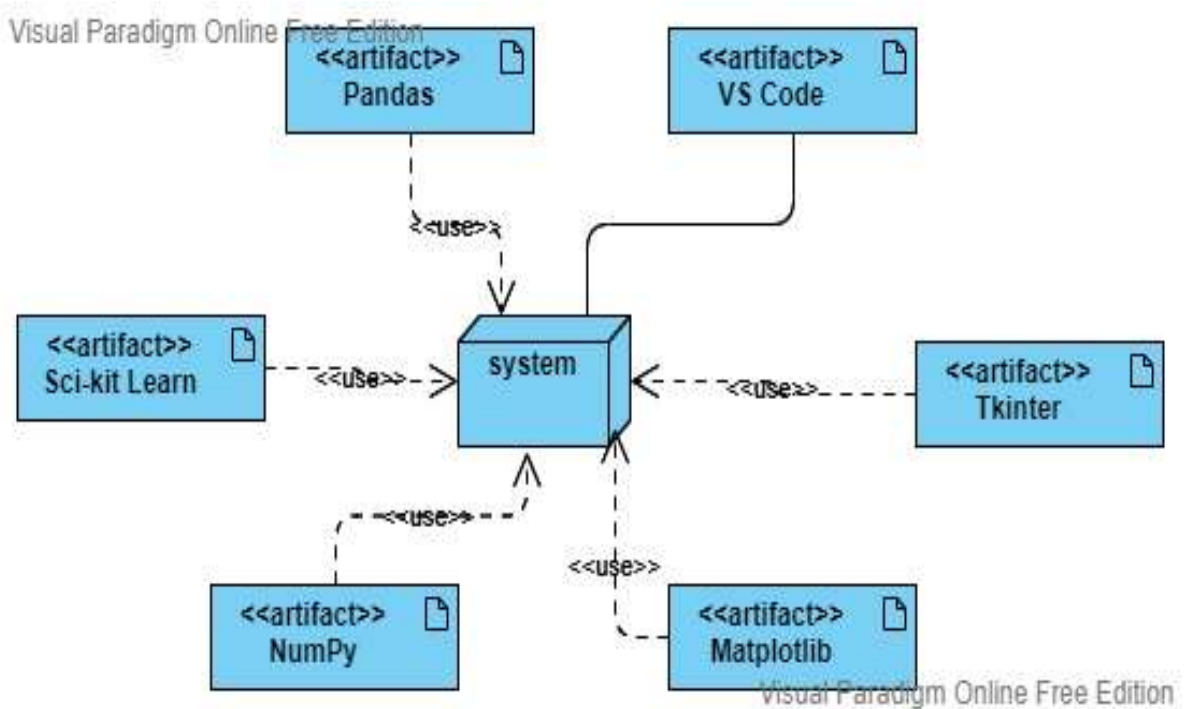
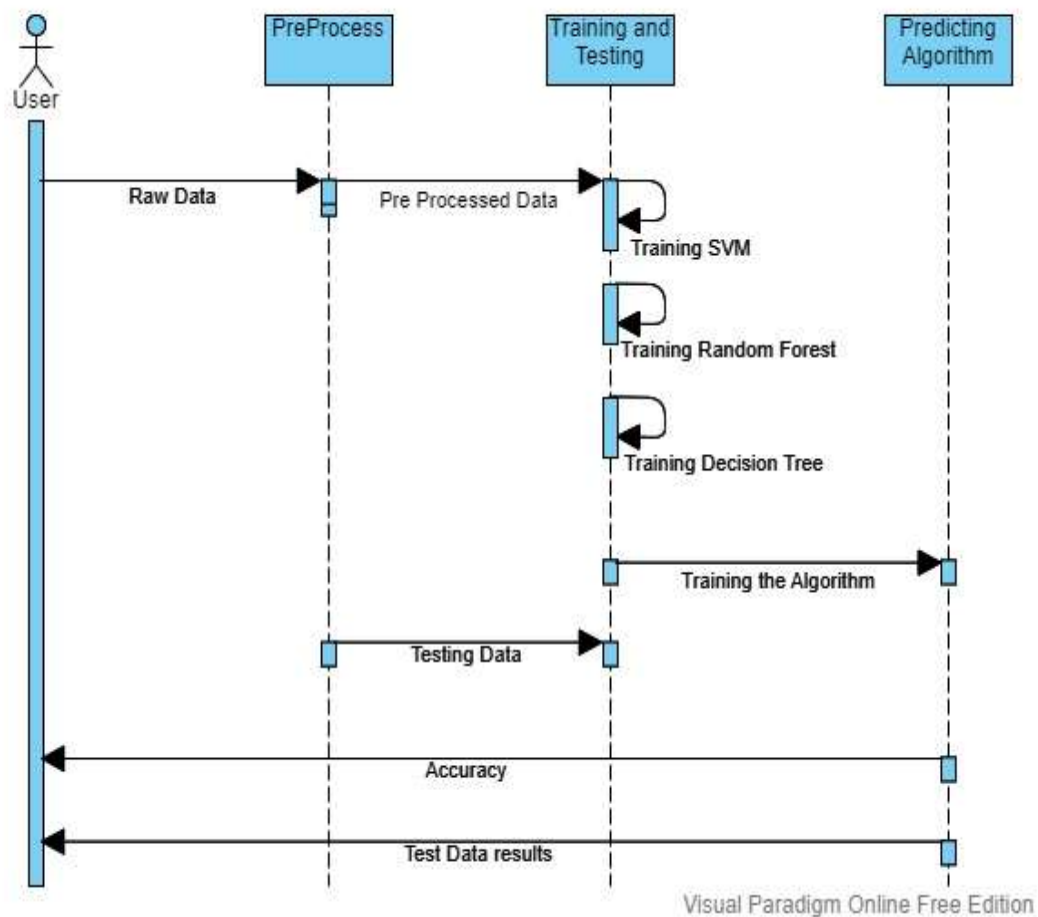


Fig 5.3.3 Deployment diagram

5.3.4 SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that emphasizes the time-ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Fig 5.3.4 Sequence diagram

5.3.5 COMPONENT DIAGRAM

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

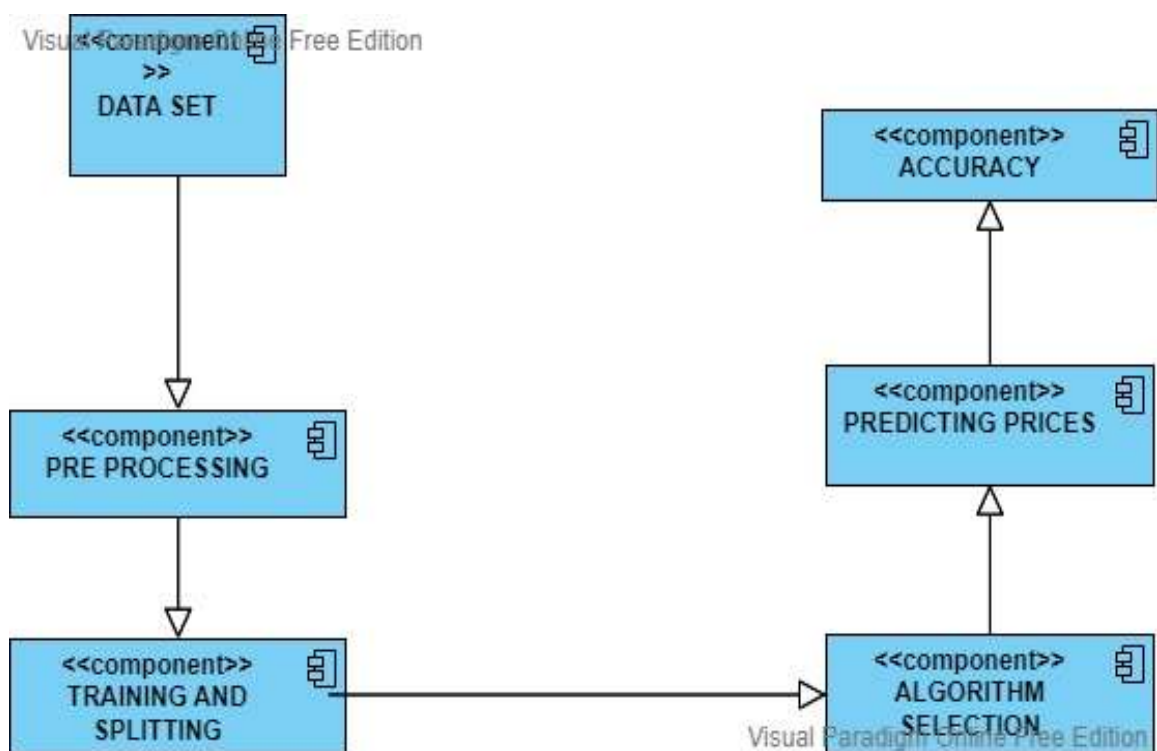


Fig 5.3.5 Component diagram

6. IMPLEMENTATION

6.1 INTRODUCTION

Software Environment:

Python language is used to implement all the required machine learning algorithms in our system.

Python

In our project, we have chosen *Python programming* language for developing the code. Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed, and garbage collected. It supports multiple programming paradigms, including structured, object-oriented, and functional programming.

The practical implementation of Python in machine learning projects and tasks has made the work easier for developers, data scientists, and machine learning engineers. Python can be easily used to analyze and compose available data, which also makes it one of the most popular languages in data science. The rich native expansion also strengthens the advantages of Python, which is more suitable for machine learning, data accounting, etc.

FEATURES OF PYTHON

High-Level Language

A high-level language (HLL) is a programming language that enables a programmer to write programs that are more or less independent of a particular type of computer. These languages are said to be high level since they are very close to human languages and far away from machine languages. Unlike C, Python is a high-level language. We can easily understand Python and it is closer to the user than middle-level languages like C. In Python, we do not need to remember system architecture or manage the memory.

Platform Independent

Platform independence is yet another amazing feature of Python. In other words, it means that if we write a program in Python, it can run on a variety of platforms, for instance, Windows, Mac, Linux, etc. We do not have to write separate Python code for different platforms.

Dynamically Typed Language

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the

6.2 Technologies Used

6.2.2 MATPLOTLIB

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

6.2.3 NUMPY:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

6.2.4 PANDAS:

Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is fast and it has high-performance & productivity for users.

6.2.5 SCI-KIT LEARN:

scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface.

Key features of scikit-learn:

- Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc.
- Accessible to everybody and reusable in various contexts.
- Built on the top of NumPy, SciPy, and matplotlib.
- Open source, commercially usable – BSD license.

6.3 Coding Standards

DMAIC is a data driven quality strategy used to improve processes. It is an integral part of six sigma initiative, but in general can be implemented as a standalone quality improvement procedure or as part of other process improvement initiatives such as lean. DMAIC is an acronym for the five phases that make up the process: • Define the problem, improvement activity for improvement, the project goals, and customer requirements. • Measure process performance • Analyze the process to determine root causes of variation, poor performance • Improve process performance by addressing and eliminating the root causes • Control the improve process and future process performance

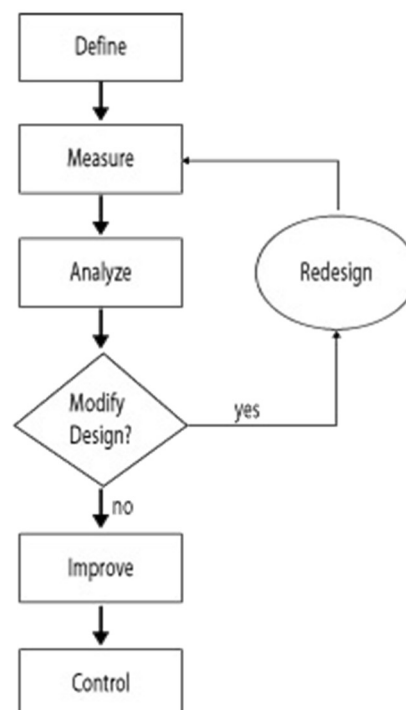


Fig 6.3 DMAIC Flow Chart

Define, measure, analyze, improve and control (DMAIC) is a structured problem-solving method. Each phase builds on the previous one, with the goal of implementing long-term solutions to problems. Sometimes, project leaders or sponsors don't feel a formal approach is necessary, but most problem-solving efforts benefit from a disciplined method. The tools used in the define phase lay the foundation for the project. The team accurately and succinctly defines the problem, identifies customers and their requirements, and determines skills and areas that need representation on the project team. Individuals who must be part of the core team or be ad-hoc members are identified, and project measures, financials and a communication plan are established. The measure phase is when the true process is identified and documented. Process steps, and corresponding inputs and outputs are identified. Measurement systems are identified or developed, and validated and improved as required. Baseline performance is established with trustworthy data. In the analyze phase, the critical inputs are identified. Inputs that have a strong relationship with the outputs and root causes are determined. These critical inputs are the drivers of performance.

In the improve phase, potential solutions are identified and evaluated, and the process is optimized. The critical inputs that must be controlled to maintain performance that reliably satisfies the customer are determined. Process capability and project financials are estimated.

The control phase establishes mistake-proof, long-term measurement and reaction plans. The team develops standard operating procedures and establishes process capability. Project financials are updated, verified and reported. Control plans are established with reaction plans, ownership and control is transitioned to the process owner, and lessons are documented. The team documents opportunities to spread the outcomes to other areas in the organization.

When to use DMAIC

When improving a current process, if the problem is complex or the risks are high, DMAIC should be the go-to method. Its discipline discourages a team from skipping crucial steps and increases the chances of a successful project, making DMAIC a process most projects should follow.

There are two approaches to implementing DMAIC. The first is the team approach in which individuals who are skilled in the tools and method, such as quality or process improvement experts, lead a team. The team members work on the project part-time while caring for their everyday responsibilities. The quality or process improvement expert might be assigned to several projects. These are long-duration projects taking months to complete.

7. SYSTEM TESTING

7.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 SOFTWARE TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2.1 TESTING METHODOLOGIES

The following are the Testing Methodologies:

- o **Unit Testing.**
- o **Integration Testing.**
- o **Functional Testing.**
- o **System Testing.**
- o **White box Testing.**
- o **Black box Testing.**

7.2.2 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge

of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.3 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.2.4 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.2.5 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test.

System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.2.6 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is used to evaluate areas that cannot be reached from a black box level.

7.2.7 Black Box Testing

Black Box Testing is evaluating the software without any knowledge of the inner workings, structure or language of the module being evaluated. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.3 TEST CASES

S. No	Test case Description	Actual value	Expected value	Result
1	Predict if Pulsar star or not	Pulsar or Not pulsar According to the input	Enter Float values only	Fail
2	Predict if Pulsar star or not	Pulsar or Not pulsar According to the input	Pulsar	True
3	Predict if Pulsar star or not	Pulsar or Not pulsar According to the input	Not Pulsar	True

Table 1: Unit Test Cases

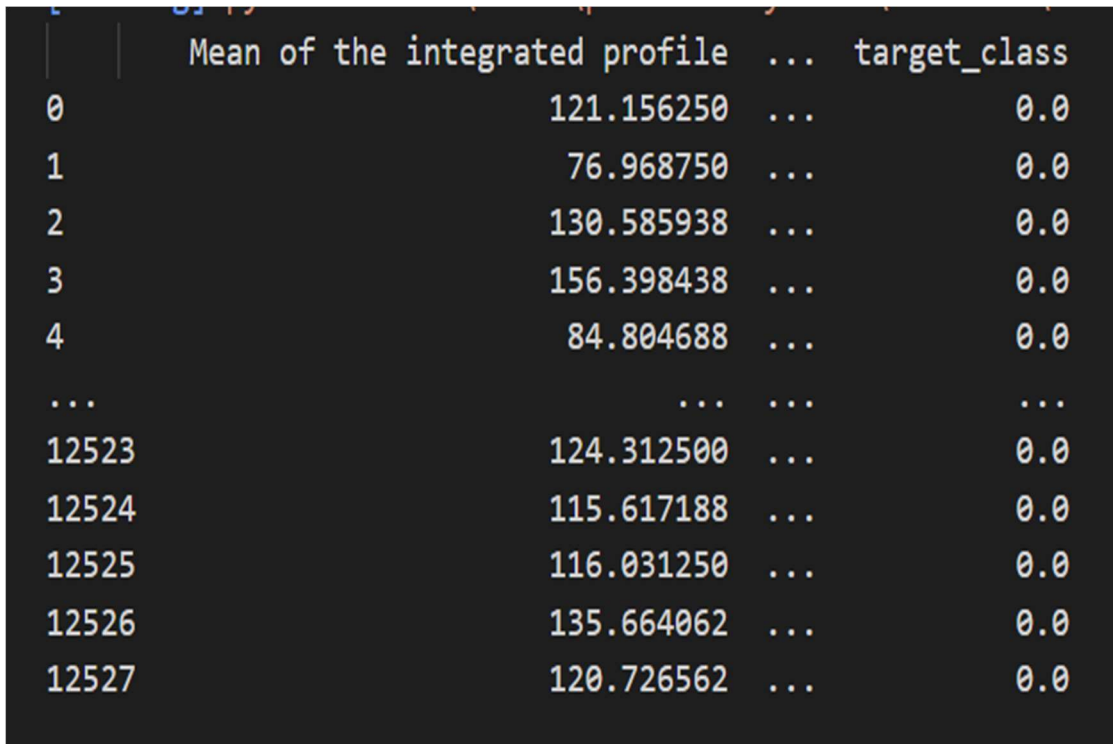
7.4 BUG REPORT

S.no	Steps	Description
1.	BugID	Nullval:log_1A-01
2.	Bug Description	Null values Entered
3.	Steps to Reproduce	Enter the correct input float values
4.	Expected output	Predicted output
5.	Actual output	Enter valid values
6.	Priority	High
7.	Severity	High

Table 2: Bug Report

8. RESULT SCREENS

Data Set



	Mean of the integrated profile	...	target_class
0	121.156250	...	0.0
1	76.968750	...	0.0
2	130.585938	...	0.0
3	156.398438	...	0.0
4	84.804688	...	0.0
...
12523	124.312500	...	0.0
12524	115.617188	...	0.0
12525	116.031250	...	0.0
12526	135.664062	...	0.0
12527	120.726562	...	0.0

Fig 8.1.1 data Set

Navigation:

- Run The python project giving the training data set as input
- It will display all the training data set data

Data Analysis

```
Data columns (total 9 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Mean of the integrated profile             12528 non-null  float64
1   Standard deviation of the integrated profile 12528 non-null  float64
2   Excess kurtosis of the integrated profile    10793 non-null  float64
3   Skewness of the integrated profile           12528 non-null  float64
4   Mean of the DM-SNR curve                    12528 non-null  float64
5   Standard deviation of the DM-SNR curve       11350 non-null  float64
6   Excess kurtosis of the DM-SNR curve          12528 non-null  float64
7   Skewness of the DM-SNR curve                 11903 non-null  float64
8   target_class                                12528 non-null  float64
dtypes: float64(9)
```

Fig 8.1.2 data Analysis

Navigation:

- Run The python project giving the training data set as input
- It will display all the training data set data

Attribute Information Graph:

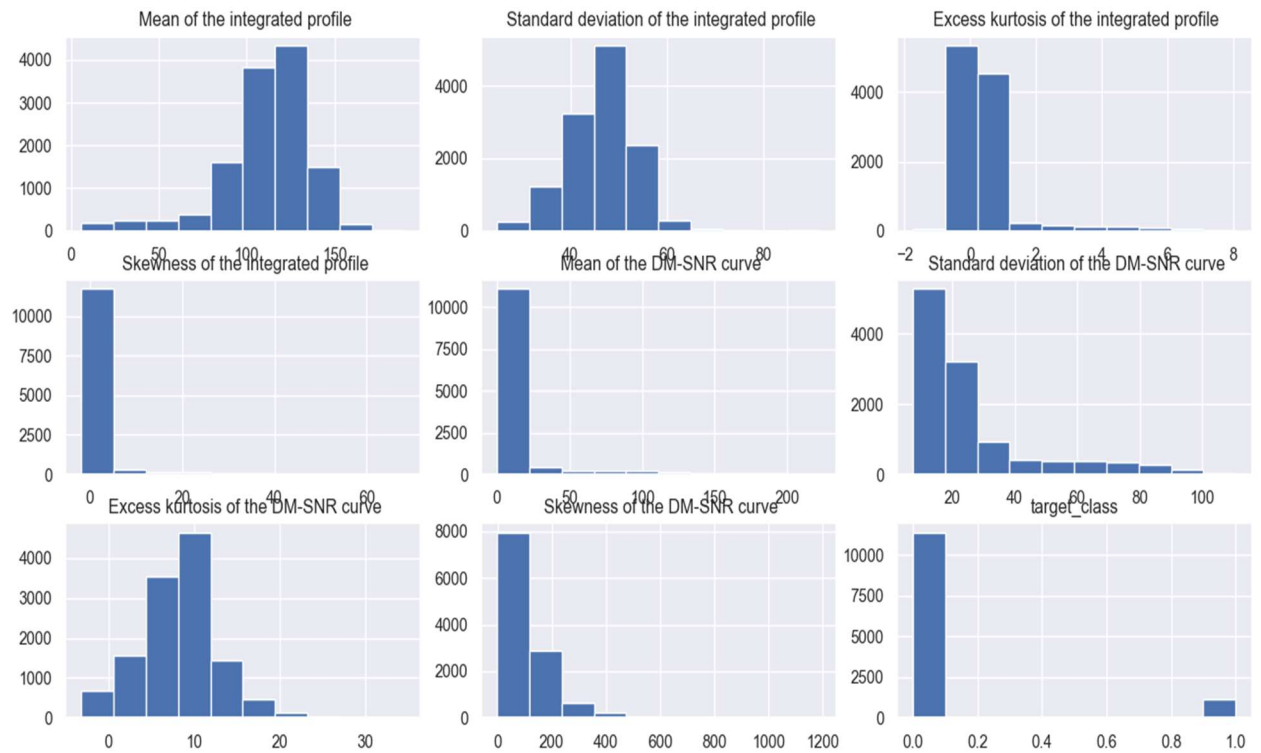


Fig 8.1.3 Attribute Information Graph

Navigation:

- All the graphs will run one after the other when you close the previous graph figure.
- Close the current graph.

Portion of Target Variables in data-set

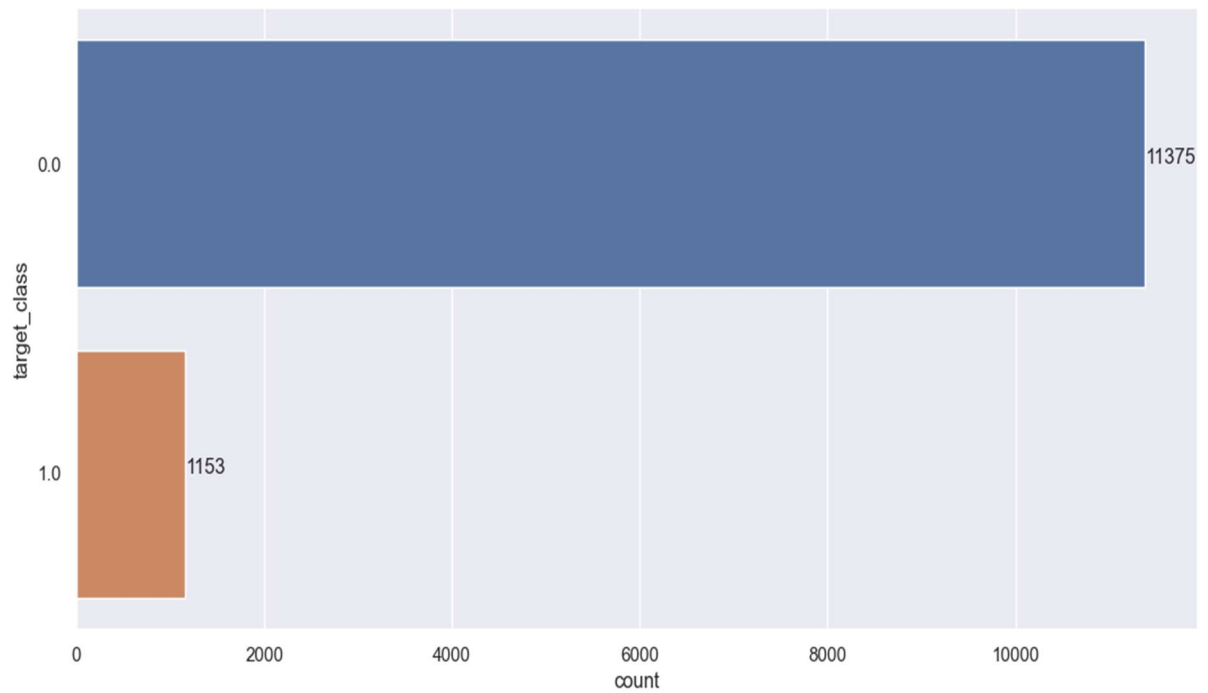


Fig 8.1.4 Portion of target variables in data set

Navigation:

- All the graphs will run one after the other when you close the previous graph figure.
- Close the current graph.

Distribution of values of each of the features

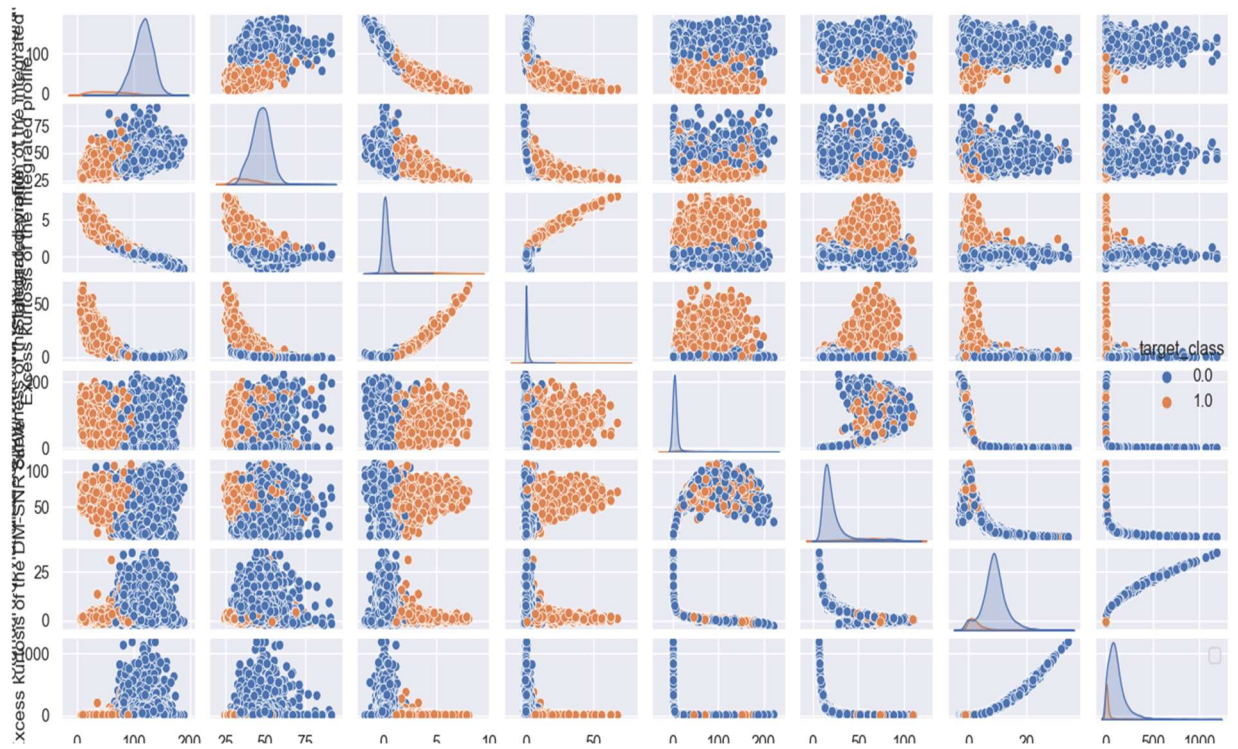


Fig 8.1.5 Distribution of values of each of the features

Navigation:

- All the graphs will run one after the other when you close the previous graph figure.
- Close the current graph.

Correlation Matrix

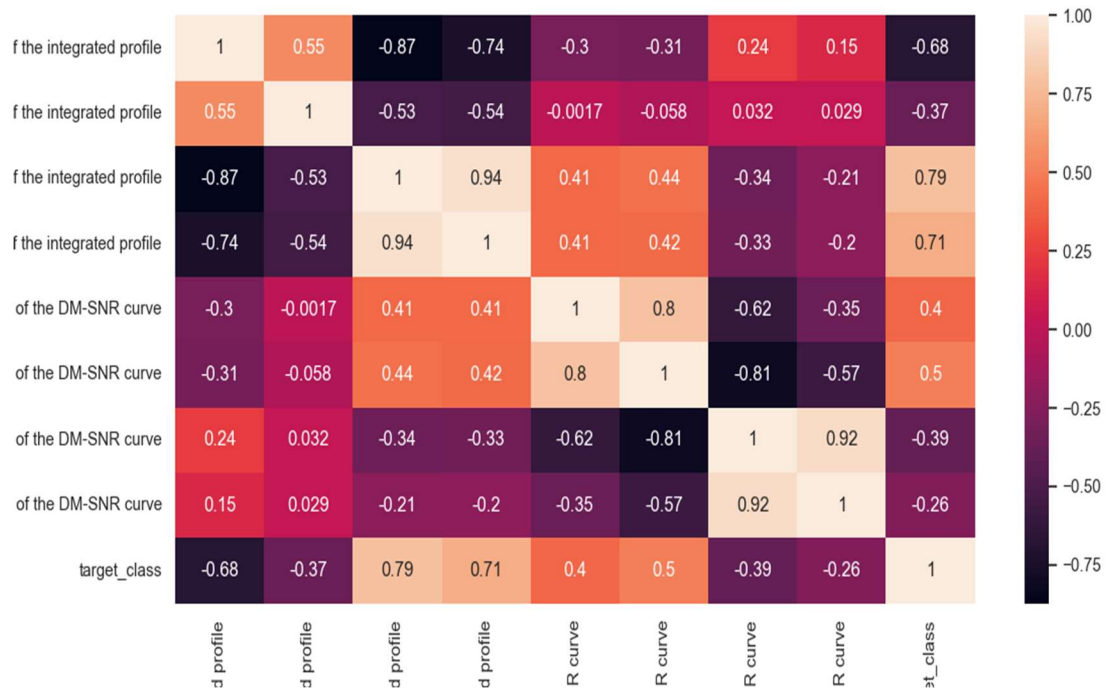


Fig 8.1.6 correlation Matrix

Navigation:

- All the graphs will run one after the other when you close the previous graph figure.
- Close the current graph.

Decision Tree Confusion matrix:

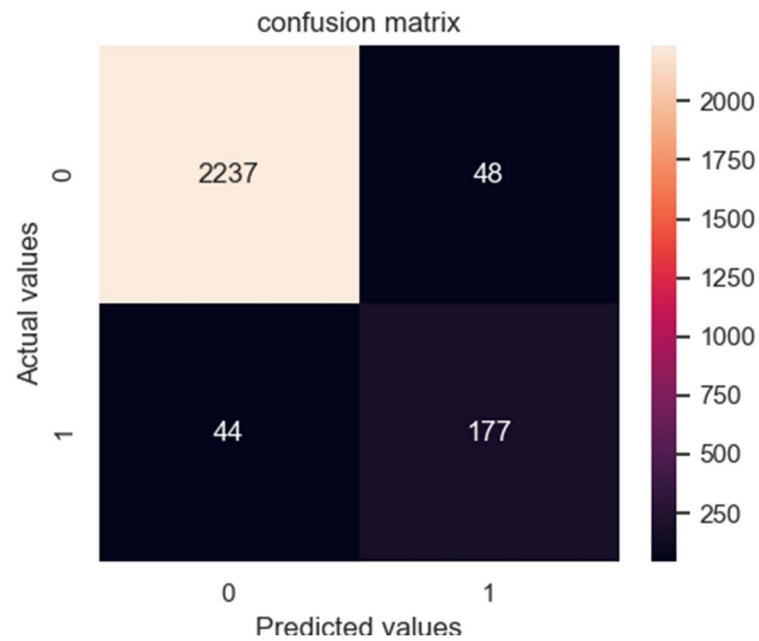


Fig 8.1.7 Confusion Matrix for Decision Tree

Navigation:

- Confusion matrix of accuracy of the decision tree algorithm will be shown along with the accuracy score based on the test data set results.
- Close the current graph to open the graph of the remaining algorithms.

Accuracy Score data of Decision Tree:

Accuracy Score: 96.36871508379889 %					
		precision	recall	f1-score	support
	0.0	0.98	0.98	0.98	2285
	1.0	0.79	0.81	0.80	221
accuracy				0.96	2506
macro avg		0.88	0.89	0.89	2506
weighted avg		0.96	0.96	0.96	2506

Fig 8.1.8 Accuracy Score Data for Decision tree

Navigation:

- Accuracy scores of the decision tree algorithm will be shown.
- Based on the accuracy of all the algorithms choose the best algorithm.

Random Forest Classifier Confusion Matrix

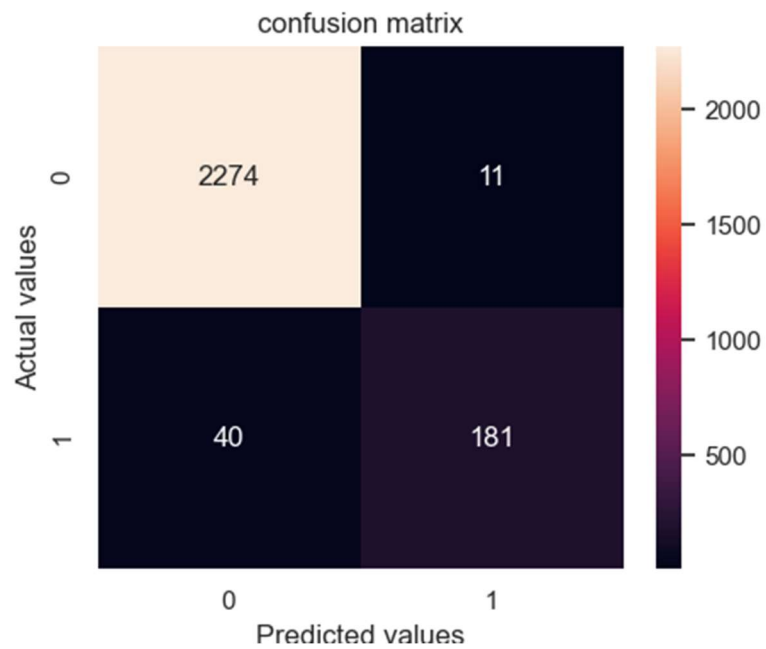


Fig 8.1.9 Confusion Matrix for random Forest Classifier

Navigation:

- Confusion matrix of accuracy of the random forest classifier algorithm will be shown along with the accuracy score based on the test data set results.
- Close the current graph to open the graph of the remaining algorithms.

Accuracy Score data of Random Forest Classifier:

Accuracy Score: 97.96488427773345 %					
		precision	recall	f1-score	support
	0.0	0.98	1.00	0.99	2285
	1.0	0.94	0.82	0.88	221
	accuracy			0.98	2506
	macro avg			0.93	2506
	weighted avg			0.98	2506

Fig 8.1.10 Accuracy Score Data for Support Vector Machine

Navigation:

- Accuracy scores of the random forest classifier algorithm will be shown.
- Based on the accuracy of all the algorithms choose the best algorithm.

Support Vector Machine Confusion Matrix:

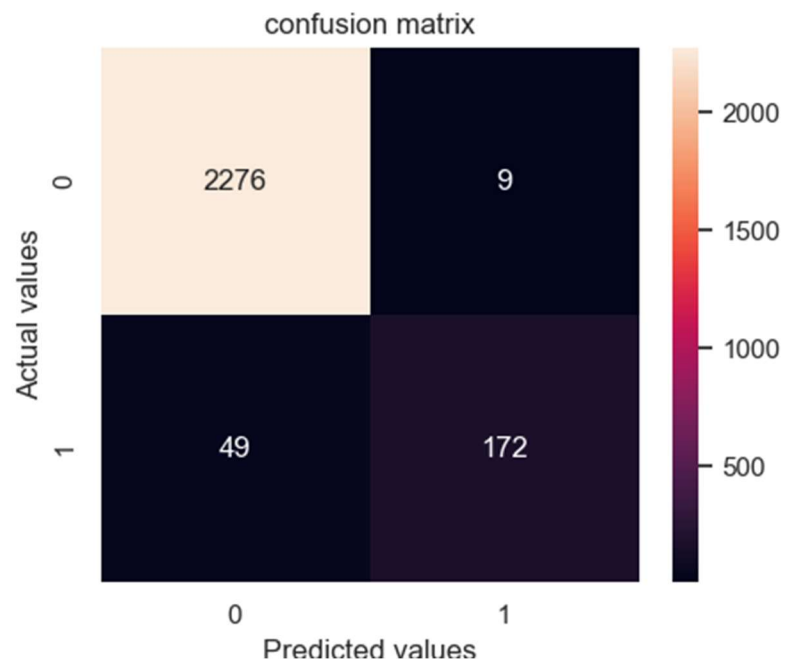


Fig 8.1.11 Confusion Matrix for Support Vector Machine

Navigation:

- Confusion matrix of accuracy of the support vector machine algorithm will be shown along with the accuracy score based on the test data set results.
- Close the current graph to open the GUI to enter star details and find out the results.

Accuracy Score data of Support Vector Machine:

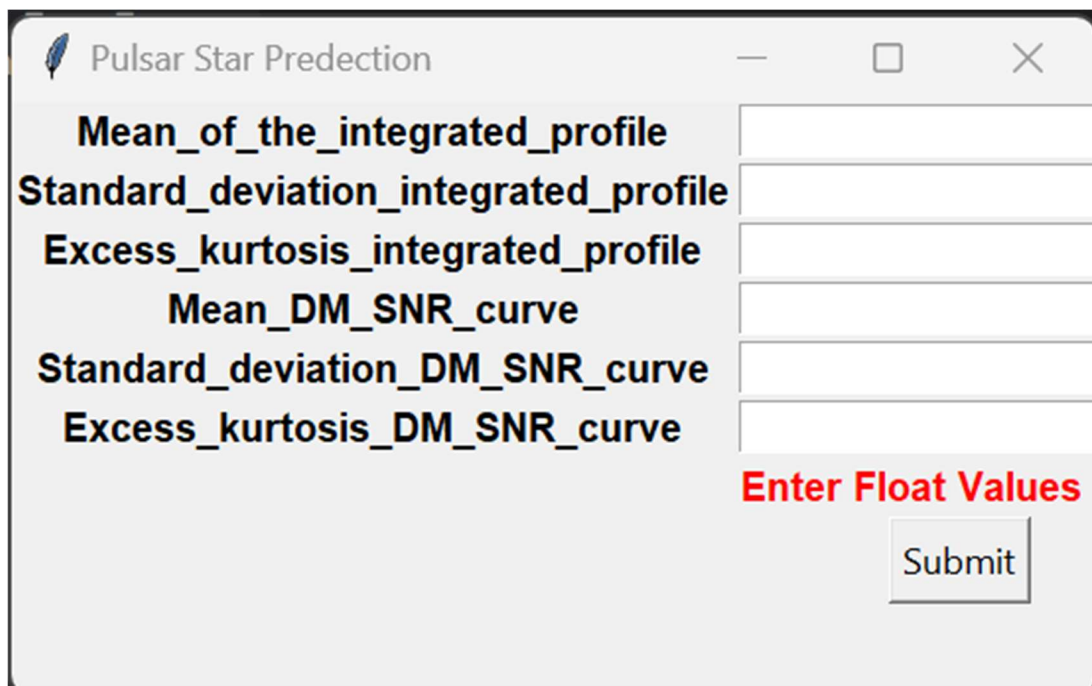
Accuracy Score: 97.6855546687949 %					
		precision	recall	f1-score	support
	0.0	0.98	1.00	0.99	2285
	1.0	0.95	0.78	0.86	221
	accuracy			0.98	2506
	macro avg			0.92	2506
	weighted avg			0.98	2506

Fig 8.1.12 Accuracy Score Data for Support Vector Machine

Navigation:

- Accuracy scores of the support vector machine algorithm will be shown.
- Based on the accuracy of all the algorithms choose the best algorithm.

Input Page



Pulsar Star Prediction

Mean_of_the_integrated_profile

Standard_deviation_integrated_profile

Excess_kurtosis_integrated_profile

Mean_DM_SNR_curve

Standard_deviation_DM_SNR_curve

Excess_kurtosis_DM_SNR_curve

Enter Float Values

Submit

Fig 8.1.13 Accuracy Score Data for Support Vector Machine

Navigation:

- After all the algorithms are trained based on the accuracy we select the best algorithm
- After closing the graphs it will automatically popup a python GUI window
- Fill the input fields with the star values that we need to predict which are shown above.
- Press submit button to find out the results.
- It will open a new window with the result.

Result Page

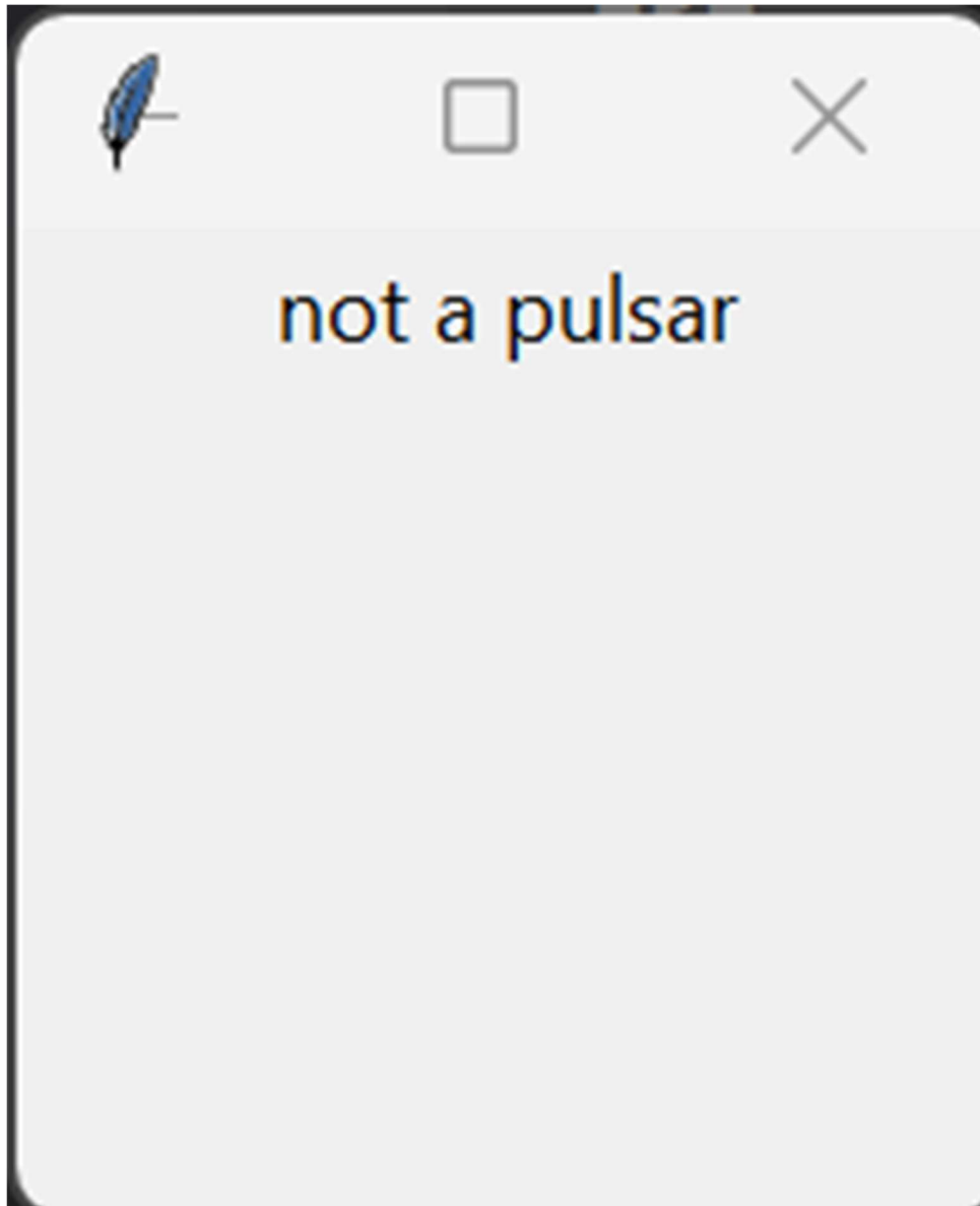


Fig 8.1.14 Accuracy Score Data for Support Vector Machine

Navigation:

- After pressing the submit button you will find out the results.
- It will open a new window with the result being “Not a Pulsar” in this case

9. CONCLUSION AND FUTURE SCOPE

CONCLUSION

In this project, we have executed 3 Machine learning models: Decision Tree Classifier, Random Forest Classifier, and Support Vector Machine to discover the pulsar star competitors. They utilize a measurable technique to produce a reasonable dataset to test the 4 models. Among these models, Random Forest Classifier plays out the best on the all-out precision while different models are just marginally more terrible than it. Our models will be helpful, and particularly Random Forest model, for future pulsar star distinguishing proof in the region of cosmology. On the off chance that pulsar stars can be discovered dependent on our forecasts, our models end up being helpful in an application.

FUTURE SCOPE

In the future, by applying different methods and techniques it can improve the accuracy score and the results can become more accurate. Test our models in more pulsar star competitor datasets other than HTRU2 to know the results accurately. Generate a superior dataset, incorporate more highlights that are in the first HTRU2 datasets, and key portrayals of Pulsars and is bigger and adjusted. Utilize better adjusting techniques to create better disseminated counterfeit information. In the event that pulsar stars can be discovered dependent on our forecasts, our AI models end up being valuable in the application.

10. BIBLIOGRAPHIES

REFERENCES

- [1] R. P. Eatough, N. Molkenhuth, M. Kramer, A. Noutsos, M. J. Keith, B. W. Stappers, and A. G. Lyne, “Selection of radio pulsar candidates using artificial neural networks,” *Monthly Notices of the Royal Astronomical Society*, vol. 407, pp. 2443–2450, 07 2010.
- [2] S. Bates, M. Bailes, B. Barsdell, N. Bhat, M. Burgay, S. Burke-Spolaor, D. Champion, P. Coster, N. D’Amico, A. Jameson, et al., “the high time resolution universe pulsar survey A Tvi. an artificial neural network and timing of 75 pulsars”, *Monthly Notices of the Royal Astronomical Society*, vol. 427, no. 2, pp. 1052-1065, 2012.
- [3] R. J. Lyon, Why are pulsars hard to find? Ph.D. thesis, The University of Manchester (United Kingdom), 2016.
- [4] Schoelkopf B, Burges C J C, Smola A J, *Advances in Kernel Methods - Support Vector Learning*. Cambridge: MIT Press, 1999.
- [5] Machine Learning Approach to Detect Pulsar Star: TensorFlow and Random Forest Model with Python
- [6] R. J. Lyon, B. Stappers, S. Cooper, J. Brooke, and J. Knowles, “Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach,” *Monthly Notices of the Royal Astronomical Society*, vol. 459, no. 1, pp. 1104–1123, 2016.
- [7] D. R. Lorimer, M. Kramer, et al., *Handbook of pulsar astronomy*, vol. 4. Cambridge university press, 2005.
- [8] R.N. Manchester and J. Taylor, “pulsars”, Freeman, San Francisco, 1977.
- [9] Ransom S., 2011, PRESTO: Pulsar Exploration and Search Toolkit. Astrophysics source code library.
- [10] Hewish A, Bell SJ, Pilkington JD, Scott PF, Collins RA. Observation of a rapidly pulsating radio source. *Nature* 1968;217(5130):709.
- [11] Bethapudi S, Desai S. Separation of pulsar signals from noise using supervised machine learning algorithms. *Astron Comput* 2018;23:15–26. 575 doi:10.1016/j.ascom.2018.02.002. arXiv:1704.04659.
- [12] Atnf pulsar catalogue. <https://www.atnf.csiro.au/research/pulsar/psrcat/>; 2019. Accessed: 2019-11-28.

- [13] Manchester RN, Hobbs GB, Teoh A, Hobbs M. The australia telescope national facility pulsar catalogue. *The Astronomical Journal* 580 2005;129(4):1993. ST. PETER'S ENGINEERING COLLEGE 56 PREDICTION OF PULSAR STAR USING MACHINE LEARNING
- [14] Johnston S, Karastergiou A. Pulsar braking and the p–diagram. *Monthly Notices of the Royal Astronomical Society* 2017;467(3):3493–9.
- [15] Lyon RJ, Stappers BW, Cooper S, Brooke JM, Knowles JD. Fifty years of pulsar candidate selection: from simple filters to 585 a new principled real-time classification approach. *Monthly Notices of the Royal Astronomical Society* 2016;459(1):1104–23. URL:<https://doi.org/10.1093/mnras/stw656>.doi:10.1093/mnras/stw656.arXiv:<http://oup.prod.sis.lan/mnras/article-pdf/459/1/1104/8115310/stw656.pdf>.
- [16] Eatough RP, Molkenhuth N, Kramer M, Noutsos A, Keith M, Stappers B, 590 et al. Selection of radio pulsar candidates using artificial neural networks. *Monthly Notices of the Royal Astronomical Society* 2010;407(4):2443–50.
- [17] Keith M, Eatough R, Lyne A, Kramer M, Possenti A, Camilo F, et al. Discovery of 28 pulsars using new techniques for sorting pulsar candidates. *Monthly Notices of the Royal Astronomical Society* 2009;395(2):837–46. 595
- [18] McLaughlin MA, Lyne A, Lorimer D, Kramer M, Faulkner A, Manchester R, et al. Transient radio bursts from rotating neutron stars. *Nature* 2006;439(7078):817.

11. APPENDIXES

11.1 SAMPLE CODE

```
# to handle tabular data
import numpy as np
import pandas as pd
# visualize the data
import seaborn as sns
import matplotlib.pyplot as plt
# Normalizing
from sklearn.preprocessing import MinMaxScaler,StandardScaler
# cross validation
from sklearn.model_selection
import train_test_split
# Algorithms
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble
import RandomForestClassifier
# Evaluation
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
import systemcheck
import warnings warnings.simplefilter("ignore")
# Data Acquisition
data = pd.read_csv("pulsar_stars.csv")
data
#Data Analysis
data.info()
data.describe()
sns.set()
data.hist(figsize = (16,10))
plt.tight_layout()
plt.show()
sns.countplot(y = data.target_class, data = data)
for index, value in enumerate(data["target_class"].value_counts()):
    plt.text(value, index, str(value))
plt.show()
plt.figure(figsize=(17,17))
sns.pairplot(data=data,hue="target_class")
plt.legend()
plt.tight_layout()
plt.show()
plt.figure(figsize=(17,17))
sns.heatmap(data.corr(),annot=True)
plt.show()
```



```

# High Correlation Filter
removable_columns = set()
corr_mat = data.corr()
for i in range(len(corr_mat.columns)):
    for j in range(i):
        if abs(corr_mat.iloc[i, j]) > 0.9:
            colname = corr_mat.columns[i]
            removable_columns.add(colname)

removable_columns
data.drop(columns=list(removable_columns),inplace=True)
data

# Normalizing the values of the Dataset
scale = StandardScaler()
scale.fit(data.drop(columns=["target_class"]))
X = scale.transform(data.drop(columns=["target_class"]))
y = data["target_class"]
X

# Data Partitioning
x_train, x_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=0)

# Decision Tree
# instantiating the Algorithm
dtc = DecisionTreeClassifier()

# Training the model
dtc.fit(x_train,y_train)

# Testing the model
dtc_pred = dtc.predict(x_test)

Evaluation
print("Accuracy Score: ",accuracy_score(y_test,dtc_pred)*100,"%")
print(classification_report(y_test,dtc_pred))
plt.figure(figsize=(5,4))
plt.title("confusion matrix")
sns.heatmap(confusion_matrix(y_test,dtc_pred),annot=True,fmt="d")
plt.xlabel("Predicted values")
plt.ylabel("Actual values")
plt.show()

# RandomForestClassifier
# instantiating the Algorithm
rfc = RandomForestClassifier(random_state=0)

# Training the model
rfc.fit(x_train,y_train)

# Testing the model
y_pred = rfc.predict(x_test)

# Evaluation
print("Accuracy Score: ",accuracy_score(y_test,y_pred)*100,"%")
print(classification_report(y_test,y_pred)) plt.figure(figsize=(5,4)) plt.title("confusion
matrix") sns.heatmap(confusion_matrix(y_test,y_pred),annot=True,fmt="d")
plt.xlabel("Predicted values")

```

```

plt.ylabel("Actual values")
plt.show()
# SVM
# instantiating the Algorithm
svm = SVC()
# Training the model
svm.fit(x_train,y_train)
# Testing the model
svm_pred = svm.predict(x_test)
# Evaluation
print("Accuracy Score: ",accuracy_score(y_test,svm_pred)*100,"%")
print(classification_report(y_test, svm_pred)) plt.figure(figsize=(5,4))
plt.title("confusion matrix")
sns.heatmap(confusion_matrix(y_test,svm_pred),annot=True,fmt="d")
plt.xlabel("Predicted values")
plt.ylabel("Actual values")
plt.show()
class_names = ["not a pulsar", "pulsar"]

def manual_pred(input_list):
    input_list = np.array(input_list)
    trans = scale.transform(input_list.reshape(1,-1))
    out = rfc.predict(trans)
    print(class_names[out[0]])

Mean_of_the_integrated_profile = 140.562500
Standard_deviation_integrated_profile = 55.683782
Excess_kurtosis_integrated_profile = -0.234571
Mean_DM_SNR_curve = 3.199833
Standard_deviation_DM_SNR_curve = 19.110426
Excess_kurtosis_DM_SNR_curve = 7.975532

manual_pred([Mean_of_the_integrated_profile,
             Standard_deviation_integrated_profile,
             Excess_kurtosis_integrated_profile,
             Mean_DM_SNR_curve,
             Standard_deviation_DM_SNR_curve,
             Excess_kurtosis_DM_SNR_curve])

```