DAYANANDA SAGAR UNIVERSITY

Devarakaggalahalli, Harohalli
Kanakapura Road, Ramanagara - 562112, Karnataka, India

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

(CUSTOMER ANALYTICS FRAMEWORK FOR CHURN MANAGEMENT IN TELCO USING BAYSIAN ANALYSIS)
Batch: 47

By

Komal Saranobat-	**ENG20CS0160**
Manje Gowda D B-	**ENG20CS0187**
Muskan A-	**ENG20CS0214**
Nayana V-	**ENG20CS0226**

Under the supervision of
Dr. Savitha Hiremath
Associate Professor, Dept. of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,

(2023-2024)

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF
ENGINEERING

Department of Computer Science & Engineering

Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Major Project Stage-II work titled **“CUSTOMER ANALYTICS FRAMEWORK FOR CHURN MANAGEMENT IN TELCO USING BAYESIAN ANALYSIS”** is carried out by **Komal Saranobat (ENG20CS0160), Manje Gowda D B (ENG20CS0187), Muskan A (ENG20CS0214), Nayana V (ENG20CS0226),** bonafide students Eighth semester of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2023-2024**.

Dr. Savitha Hiremath

Associate Professor
Dept. of CS&E,
School of Engineering
Dayananda Sagar University

Date:

Dr. Girisha G S

Chairman CSE
School of Engineering
Dayananda Sagar University

Date:

Dr. Udaya Kumar Reddy K R
Dean
School of Engineering
Dayananda Sagar University

Date:

Name of the Examiner **Signature of Examiner**

1.

2.

DECLARATION

We, **Komal Saranobat (ENG20CS0160), Manje Gowda D B (ENG20CS0187), Muskan A (ENG20CS0214), Nayana V (ENG20CS0226),** are students of eighth semester B. Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major Project Stage-II titled **"Customer analytics framework for Churn management in telco using Bayesian analysis"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2023-2024.**

Student	**Signature**
Name: Komal Saranobat **USN: ENG20CS0160**	
Name: Manje Gowda D B **USN: ENG20CS0187**	
Name: Muskan A **USN: ENG20CS0214**	
Name: Nayana V **USN: ENG20CS0226**	

Place: Bangalore

Date:

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank ***Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University*** *for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. Girisha G S, Department Chairman****,* ***Computer Science and Engineering****,* ***Dayananda Sagar University,*** *for providing right academic guidance that made our task possible.*

We would like to thank our guide ***Dr. Savitha Hiremath****,* ***Associate Professor****,* ***Dept. of Computer Science and Engineering****,* ***Dayananda Sagar University****, for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our ***Project Coordinator Dr. Meenakshi Malhotra*** *and* ***Prof. Mohammed Khurram J*** *as well as all the staff members of Computer Science and Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

NOMENCLATURE USED

ML	Machine Learning
KNN	K-Nearest Neighbour
SMOTE	Synthetic Minority Oversampling Technique
BLR	Bayesian Logistic Regression
GNB	Gaussian Naive Bayes
ANN	Artificial Neural Networks
MNB	Multinominal Naïve Bayes
BNB	Bernoulli Naïve Bayes

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

ABSTRACT

The research presents a comprehensive Customer Analytics Framework for Churn Management in the telecommunications sector, leveraging Bayesian analysis. In the dynamic landscape of telecommunication, where customer retention is paramount, the framework integrates sophisticated Bayesian models to predict and manage customer churn effectively. The proposed framework consists of six key components: data preprocessing, exploratory data analysis (EDA), churn prediction, factor analysis, customer segmentation, and customer behavior analytics. The Bayesian approach allows for probabilistic modeling, accommodating the inherent uncertainties in customer behaviors and market dynamics.

The framework's predictive capabilities are enhanced through the utilization of machine learning classifiers, including Logistic Regression, Decision Tree, K-Nearest Neighbor, Random Forest, and more. Performance metrics such as accuracy, recall, and F1-score are employed for model evaluation. Bayesian Logistic Regression is then applied to conduct factor analysis, identifying crucial features for customer segmentation. The segmentation process utilizes K-means clustering to categorize customers into distinct groups, facilitating targeted retention strategies.

Through experimentation and evaluation on diverse datasets, the framework aims to offer telecom operators a robust toolset for proactive churn management. By combining Bayesian analysis with machine learning and customer segmentation, the proposed framework strives to provide a holistic solution, empowering telecommunication companies to optimize resource allocation, foster long-term customer relationships, and ultimately mitigate the impact of customer churn on them

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

In the rapidly evolving telecommunications sector, keeping customers is key to long-term prosperity. Losing customers, also known as churn, can greatly affect earnings and profitability. To combat this issue, a strong Customer Analytics Framework is essential. This framework utilizes Bayesian Analysis, a potent statistical tool, to understand customer actions and anticipate possible churn.

1.1. UNDERSTANDING CHURN IN TELECOM

Predicting customer churn in a telecom business is vital for keeping customers and increasing profits. Bayesian analysis is a statistical technique that can help in predicting churn by incorporating prior knowledge and updating it with new data.

1.1.1. CUSTOMER SEGMENTATION AND CHURN PREDICTION

Customer segmentation is an important strategy where customers are grouped based on their characteristics and behaviors that affect churn. Predicting which customers are likely to leave a telecom business is crucial for keeping customers and increasing revenue. Bayesian analysis is a statistical technique that can be used for predicting churn by combining prior knowledge with new data.

- *Retention Methodologies:*

In today's competitive market, keeping customers coming back is key for any business. By focusing on strategies to retain existing customers, such as providing top-notch customer service and consistently meeting expectations, companies can build strong customer loyalty. This not only helps reduce costs compared to constantly acquiring new customers but also helps decrease customer churn rates in the long run.

1.2. FIGURES AND TABLES

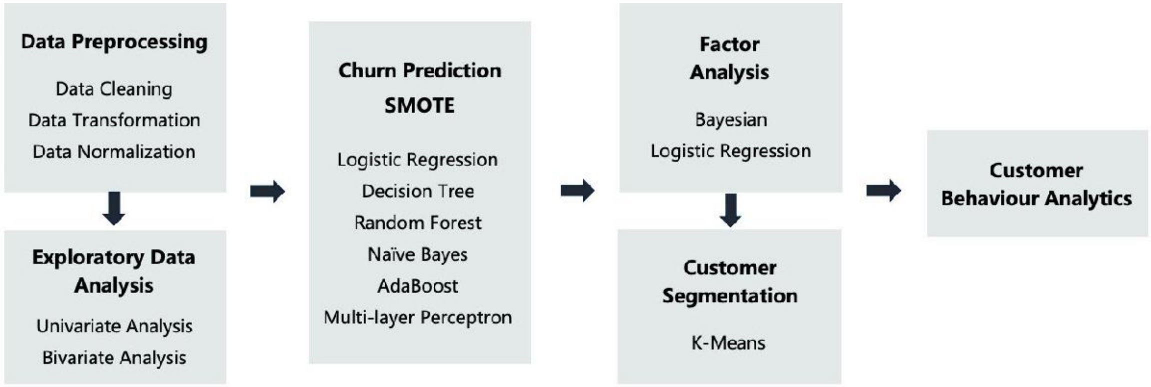
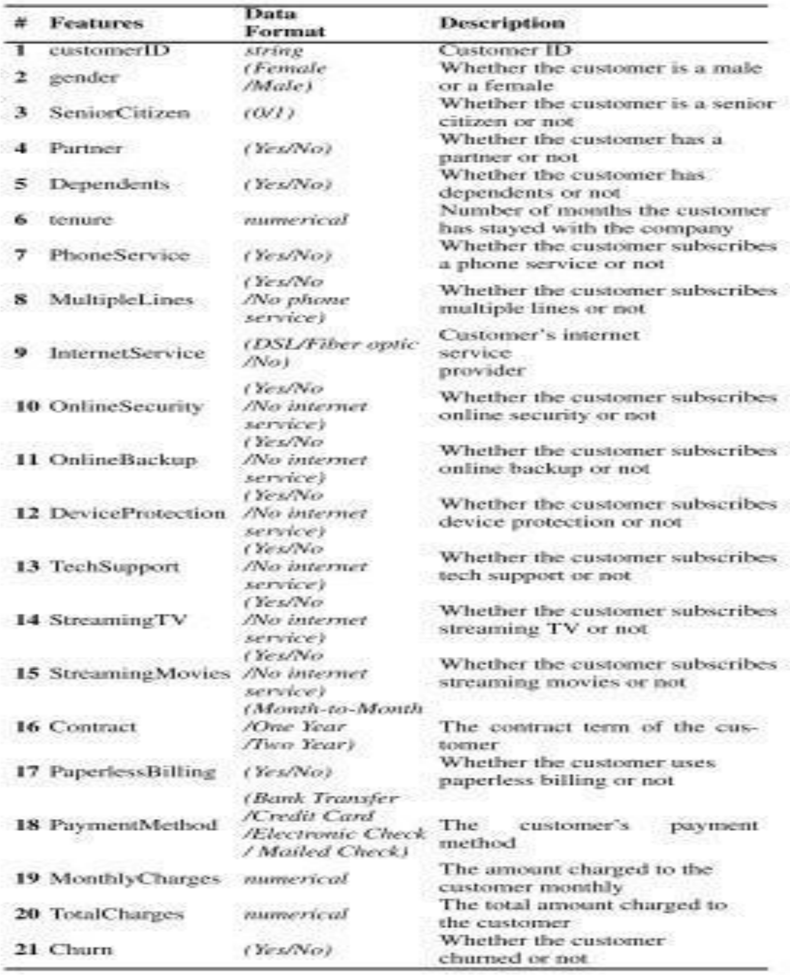


Figure 1.2 Integrated telco customer analytics framework

Figure 1.2 shows a data preparation for customer churn prediction and segmentation in the telecommunication industries.

Table 1.2: Features in Dataset



#	Features	Data Format	Description
1	customerID	*string*	Customer ID
2	gender	*(Female /Male)*	Whether the customer is a male or a female
3	SeniorCitizen	*(0/1)*	Whether the customer is a senior citizen or not
4	Partner	*(Yes/No)*	Whether the customer has a partner or not
5	Dependents	*(Yes/No)*	Whether the customer has dependents or not
6	tenure	*numerical*	Number of months the customer has stayed with the company
7	PhoneService	*(Yes/No)*	Whether the customer subscribes a phone service or not
8	MultipleLines	*(Yes/No /No phone service)*	Whether the customer subscribes multiple lines or not
9	InternetService	*(DSL/Fiber optic /No)*	Customer's internet service provider
10	OnlineSecurity	*(Yes/No /No internet service)*	Whether the customer subscribes online security or not
11	OnlineBackup	*(Yes/No /No internet service)*	Whether the customer subscribes online backup or not
12	DeviceProtection	*(Yes/No /No internet service)*	Whether the customer subscribes device protection or not
13	TechSupport	*(Yes/No /No internet service)*	Whether the customer subscribes tech support or not
14	StreamingTV	*(Yes/No /No internet service)*	Whether the customer subscribes streaming TV or not
15	StreamingMovies	*(Yes/No /No internet service)*	Whether the customer subscribes streaming movies or not
16	Contract	*(Month-to-Month /One Year /Two Year)*	The contract term of the customer
17	PaperlessBilling	*(Yes/No)*	Whether the customer uses paperless billing or not
18	PaymentMethod	*(Bank Transfer /Credit Card /Electronic Check / Mailed Check)*	The customer's payment method
19	MonthlyCharges	*numerical*	The amount charged to the customer monthly
20	TotalCharges	*numerical*	The total amount charged to the customer
21	Churn	*(Yes/No)*	Whether the customer churned or not

Table 1.2 shows features for a customer dataset. It includes 21 features such as customer ID, gender, senior citizen, churn etc.

1.3. SCOPE

In the telecommunications industry, implementing a Customer Analytics Framework for Churn Management using Bayesian analysis shows great potential. This method helps telecom companies improve customer retention strategies through probabilistic modeling. Bayesian analysis allows for a flexible churn prediction system that adapts to changing customer behaviors and market trends. The framework also includes real-time analysis, giving timely information on customer churn probabilities.

By using Bayesian methods, the model can constantly learn and improve with new data, ensuring its continued relevance and accuracy. The system is able to recognize small trends and connections in customer information, allowing for personalized strategies to keep customers who may be at risk of leaving. Furthermore, Bayesian analysis helps account for uncertainties in the telecom industry, like changes in network quality and competition. The system is flexible and can handle large amounts of data, making it ideal for telecommunications companies.

By using Bayesian analysis for Customer Analytics Framework for Churn Management, telecom providers can prevent customers from leaving, allocate resources effectively, and build lasting relationships.

Social impact:

When implementing a customer analytics framework, it is important to prioritize data privacy and security. By identifying and resolving issues that cause customers to switch to a different provider, telecom companies can improve the overall customer experience.

Using customer analytics to optimize services may result in more affordable and easily accessible telecom options, as well as contribute to the sustainability and expansion of the telecom industry. Upholding ethical standards in data management and analysis can help build trust with users and the community.

Technical Impact:

This project has a significant impact on data science, analytics, and information technology. To succeed, it needs a well-integrated and technically advanced approach to fully utilize customer analytics for managing churn in the telecommunications industry.

CHAPTER 2
PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

Problem: One of the biggest issues that telecommunication companies face is customer churn, where customers switch to other providers or cancel services. This not only results in revenue loss but also impacts long-term growth and customer satisfaction.

CHAPTER 3
LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

3.1. Minimization of Churn Rate Through Analysis of Machine Learning.

Authors: Chaithra K N, Manu M N, Shrikanth N G, Anupama K, Soundarya B C, Gururaj H L.

Publication: International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), 29 April 2023

Customer attrition, commonly referred to as customer churn, presents a significant issue within the banking industry, leading to around 1.5 million customers departing each year. Factors such as rival financial offerings, branch accessibility, and interest rates influence customers' choices to switch banks. To combat this trend, predictive models like Logistic Regression, Decision Tree, K-Nearest Neighbor, and Random Forest are utilized. Through assessing their accuracy metrics, it is evident that Random Forest stands out with a 71% accuracy rate, showcasing its ability to reduce customer attrition and enhance profitability.[1]

3.2. Integration of Churn Predictions and the Customer Segmentation Framework for the Telco Business.

Authors: SHULI WU, WEI-CHUEN YAU, THIAN-SONG ONG, and SIEW-CHIN CHONG are members of the IEEE. They are affiliated with the School of Electrical and Computer Engineering at Xiamen University Malaysia, located in Sepang 4390.

Publication: April 16, 2021,

In the telecom industry, this study highlights the importance of focusing on retaining existing customers rather than solely acquiring new ones. The framework proposed integrates with churn prediction and customer segmentation, which helps in the telco operators effectively manage customer churn. Machine learning classifiers, such as AdaBoost for Dataset 1, Random Forest for Dataset 2, and Bayesian Logistic Regression for factor analysis, were utilized with success. The K-means clustering which was then used to segment the customers, leading to improved targeted retention strategies. This comprehensive strategy guarantees precise churn analysis and optimization of customer management tactics.[2]

3.3. An efficient system for the customer churn predictions through the particle swarm optimization-based on feature selection model with simulated annealing.

Authors: J. Vijaya1 · E. Sivasankar1

Publication: Springer Science+Business Media, LLC, 19 September 2017

In this article, we discuss the issue of forecasting customer turnover in the telecommunications industry, which is important because of the concerns of stakeholders, fierce competition, and significant financial losses. We introduce three different versions of a model using particle swarm optimization (PSO), which include feature selection, simulated annealing, and a blend of the two. Our comparison analysis demonstrates that the PSO-based models outperform traditional classifiers in terms of effectiveness and accuracy.[3]

3.4. Churn Prediction in the Telecom Business.

Authors: Georgina Esteves, Joao Mendes-Moreira

Publication: The eleventh international conference on digital information management (ICDIM) 2016

Telecommunication companies understand the connection between customer happiness and profits. A research conducted using actual data from WeDo Technologies evaluated six different algorithms (Algorithm KNN, Naive Bayes, C4.5, Random Forest, AdaBoost, and the ANN) to forecast customer turnover. The study found that Random Forest was the most successful algorithm across all measures, highlighting its practicality for real-world use.[4]

3.5. Enhanced Churn Predictions in the Telecommunication Industry.

Authors: Written by Awodele Oludele, Adeniyi Ben*, Ogbonna A.C., Kuyoro S.O., and Ebiesuwa Seun, a research paper was published in the International Journal of Innovative Research in Computer Science & Technology (IJIRCST) in March 2020.

In order to anticipate customer defection, a Markov Chain Model is utilized, which enables adaptability and takes into account fluctuating retention rates. MATLAB Monte Carlo simulations are used to calculate churn rates, establishing a reference churn value for each customer. The Markov model streamlines the evaluation of dynamic decisions, leading to a comprehensive grasp of customer relationships and probabilities.[5]

3.6. The review on the Churn Predictions and the Customer Segmentation using Machine Learning.

Authors: written by Ankitha Zadoo, Tanmay Jagtap, Nikhil Khule, Ashutosh Kedari, Shilpa Khedkar.

Publication: The International Conference on Machine Learning, Big Data, Cloud, and Parallel Computing (COM-IT-CON) happening on the 26th and 27th of May 2022.

Telecommunication companies use a large amount of customer data to analyze behavior and plan for customer retention. By segmenting customers through methods such as data cleaning and feature selection using techniques like PCA and LDA, companies are better able to understand and anticipate customer churn. This research delves into methods for predicting churn and segmenting customers to create more effective models.[6]

3.7. The Churn predictions on the huge telecom data using the hybrid firefly based classification.

Authors: Amar A. Q. Ahmed, Maheshwari D.

Publication: Egyptian Informatics Journal, 1 March 2017

In this study, we explore the challenges of predicting customer churn in the telecommunications industry by utilizing a metaheuristic approach that combines elements of a Firefly algorithm with Simulated Annealing. By replacing the computationally intensive tasks with Simulated Annealing, we were able to improve efficiency. Our experiments with the Orange dataset show that this hybrid Firefly algorithm is not only effective but also faster in predicting churn.[7]

3.8. Comparing the Oversampling Techniques to handle a Class Imbalance Problem.

Authors: Adana Amin, Sajid anwar, Awais Adanan, Muhammad Nawaz, Newton Howard, Junaid Qadir (Senior Member, IEEE), Ahmad Hawalah, and Amir Hussain (Senior Member, IEEE).

Publication: October 26, 2016

In industries that provide services, particularly in the telecom sector, maintaining customer loyalty depends on predictive models that are influenced by uneven datasets. This study examines and contrasts six sampling approaches, such as MTDF, that aim to tackle issues related to imbalanced classes. Findings indicate that MTDF and genetic algorithms surpass other techniques in terms of predictive accuracy and the creation of rules.[8]

3.9. The survey of the Predictive Modelling under the Imbalanced Distributions.

Authors: Paula Branco, Luis Torgo, Rita P. Ribeiro

Publication: Association for Computing Machinery New York,13 August 2016

In this article, we explore methods for dealing with imbalanced data in practical data-mining scenarios. This is especially important when dealing with infrequent values in the target variable, such as in fraud detection. We discuss the difficulties that arise, suggest a problem statement, and outline different approaches for both classification and regression tasks. Our investigation includes discussions on techniques, classifications, comparisons, theoretical examinations, and other predictive modeling challenges, offering a broad overview of ways to handle imbalanced datasets.[9]

3.10. Just in time , the customer churn predictions in the telecommunication sectors.

Authors: Adnan Amin, Feras AI obeidat Babar Shah, May Al Tae, Changez Khan, Hamood- Rehaman Durrani, Sajid Anwar

Publication: The Journal of Supercomputing · June 2020

The telecom industry is facing significant customer churn challenges due to rapid growth and fierce competition. This research aims to tackle the problem of limited historical data in predicting customer churn by introducing a Just-in-Time approach that utilizes data from multiple companies. Results from analyzing data from two telecom companies demonstrate that a diverse which ensemble-based Just-in-Time the model for the Customer Churn Predictions is more effective than using individual classifiers or uniform ensemble methods. This offers a valuable solution for companies that do not have extensive historical data..[10]

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

4.1. System architecture

When designing a system architecture for a web application, it is important to consider the overall hypermedia structure. The architecture should be aligned with the goals of the web application, the type of content it will contain, the target users, the navigation approach which has been decided upon. The content architecture involves organizing content objects for presentation and navigation. Web application architecture deals with how these applications are designed to handle the user interaction, the internal processing, navigation, and content presentation. The architecture of a web application is determined by the development environment in which it will be implemented.

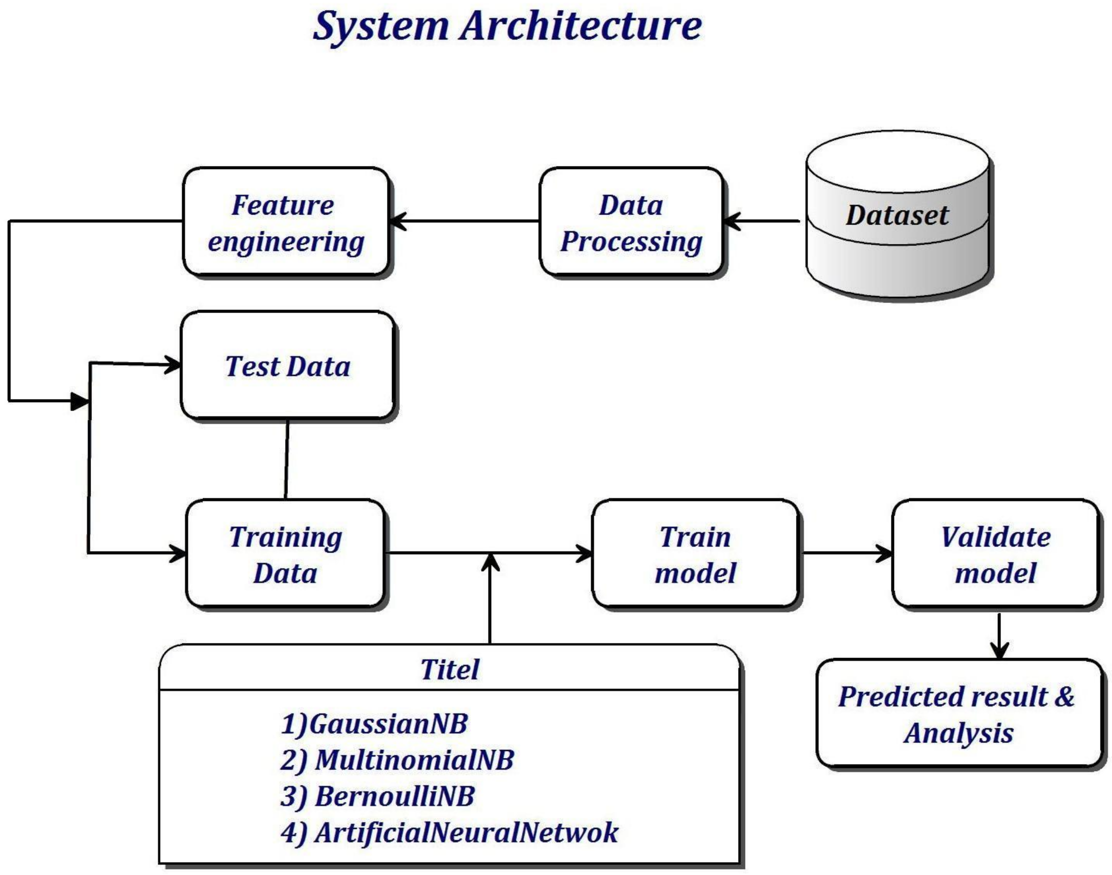


Figure 4.1. System architecture

Figure 4.1 shows the different stages involved in training a machine learning model, including data processing, training data, test data, validating the model, and analyzing the predicted results. The diagram also shows four different machine learning models that can be used for this process.

4.2. Flow chart diagram

It's crucial to finish all tasks on time. There are several project management tools that can assist project managers in organizing their tasks and schedule, such as this flowchart. The flowchart is one of the seven fundamentals of quality tools which utilized in the project management, showing the necessary actions for to achieve the objectives of a specific task in a practical order. Also known as the process maps, this tool illustrates a series of steps with branching options that shows inputs being transformed into the outputs. The benefits of the flowcharts is that they outline the project's activitie, including decision points, parallel paths, branching loops, and the overall sequence.

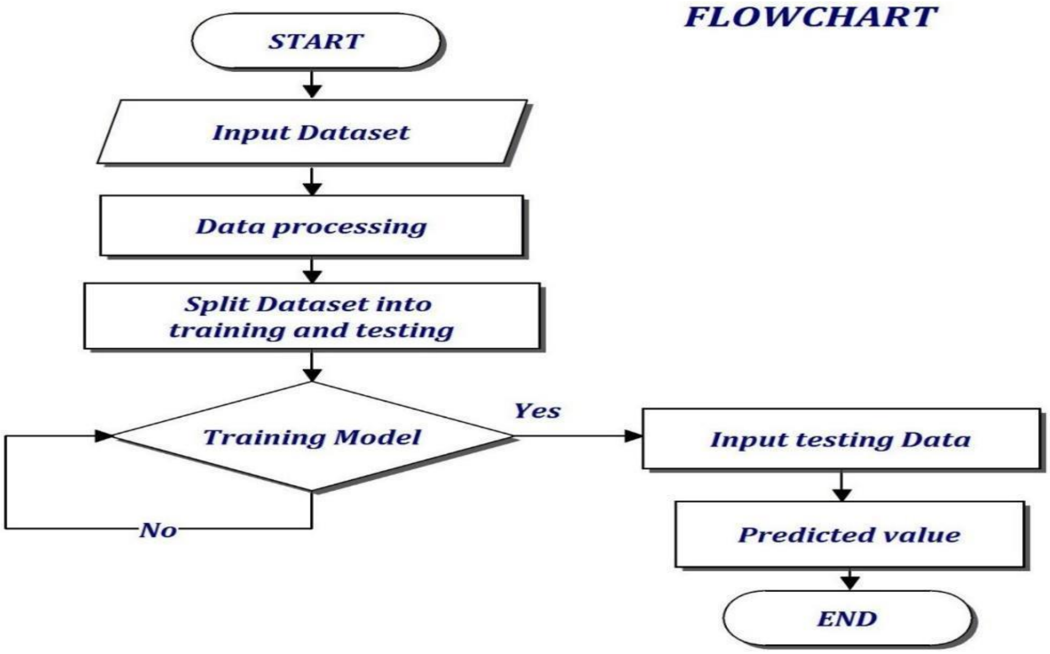


Fig 4.2 Flow Chart Diagram

Figure 4.2 The flowchart shows how data is collected, preprocessed, and which then this divided into the training and the testing datasets. The model which is trained using the the training data and is then evaluated using the testing data.

4.3 Use case diagrams

A use case is a series of scenarios that explain how a source and a destination interact. A use case diagram shows how actors and use cases are related. The main key element of the use case diagrams are the use cases and actors. The diagram illustrates the interaction between them.

4.3.1 Use case diagram for user

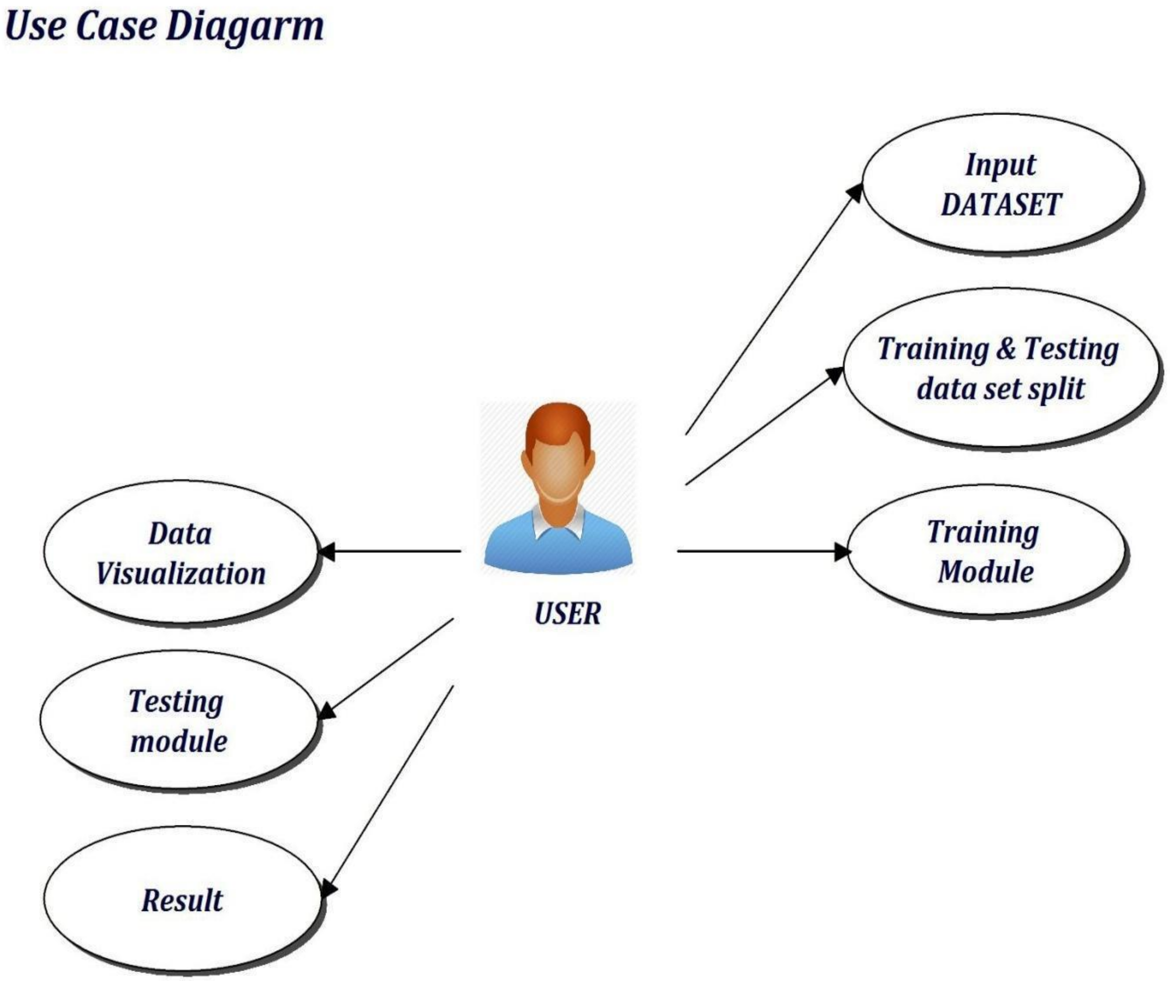


Fig 4.3.1 Use Case Diagram for User

In Figure 4.3.1, we see a diagram illustrating the process of splitting a training and testing dataset. The diagram demonstrates how a user interacts with a system to input a dataset and then splits it into training and testing data for analysis.

4.4. Data flow diagram

1. A data flow diagram (DFD) is a visual representation of how data moves through a system. It can also be used to show how data is processed. Designers typically start with a context level DFD, which illustrates how the system interacts with external entities. DFDs show how data flows into the system from external sources, moves between processes, and is stored logically. There are four symbols used in DFDs.

2. Squares represent external entities, which are where information comes into and goes out of the system.

3. The other methodologies, processes which may be which referred to as 'Activitie', 'Action', 'Procedure', 'Subsystem', etc. These processes take data as input, perform operations on this, and then produce the output.

4. Data flows are depicted by arrows, carrying either electronic information or physical objects. Data cannot move between storage locations without passing through a designated process, and external entities do not have direct access to these storage locations.

5. The rectangular shape, resembling a flat three-sided object, serves the purpose of managing data storage by receiving and distributing information for processing purposes.

4.4.1 Level 0 dataflow diagram

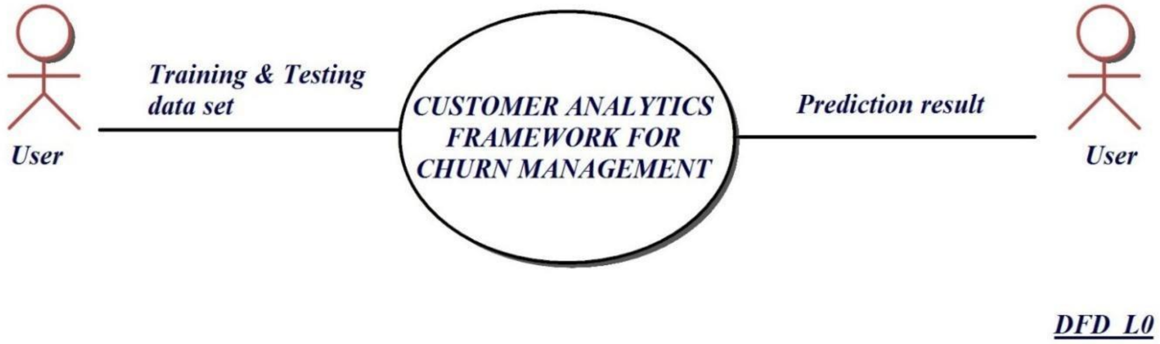

User
Training & Testing data set
CUSTOMER ANALYTICS FRAMEWORK FOR CHURN MANAGEMENT
Prediction result
User
DFD_L0

Fig 4.4.1 Level 0 Dataflow diagram

Figure 4.4.1 discusses a customer analytics framework for managing churn. It describes a simple three-step process which involves creating training and testing datasets, implementing the customer analytics framework, and analyzing the prediction results.

4.4.2 Level 1 dataflow diagram

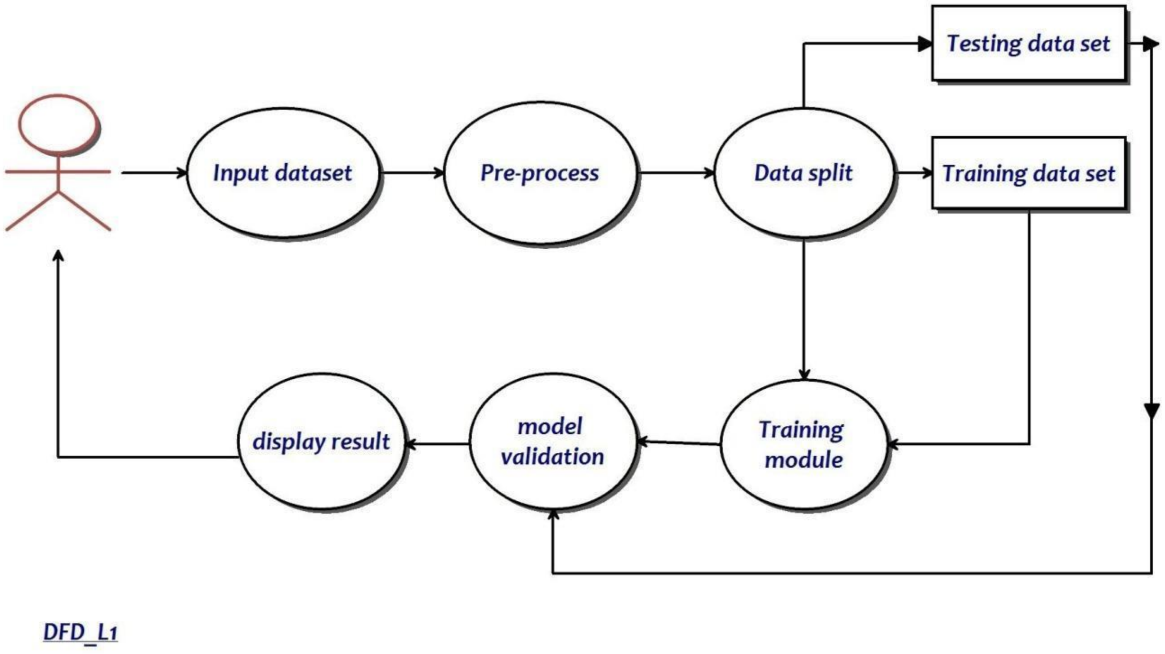


Fig 4.4.2 Level 1 Dataflow Diagram

Figure 4.4.2, you can see a flow diagram depicting how a model is trained to detect autism. The data goes through preprocessing, is divided into the training and the testing sets, and then where the model is trained using this data. After that, the model's accuracy is assessed on the testing set.

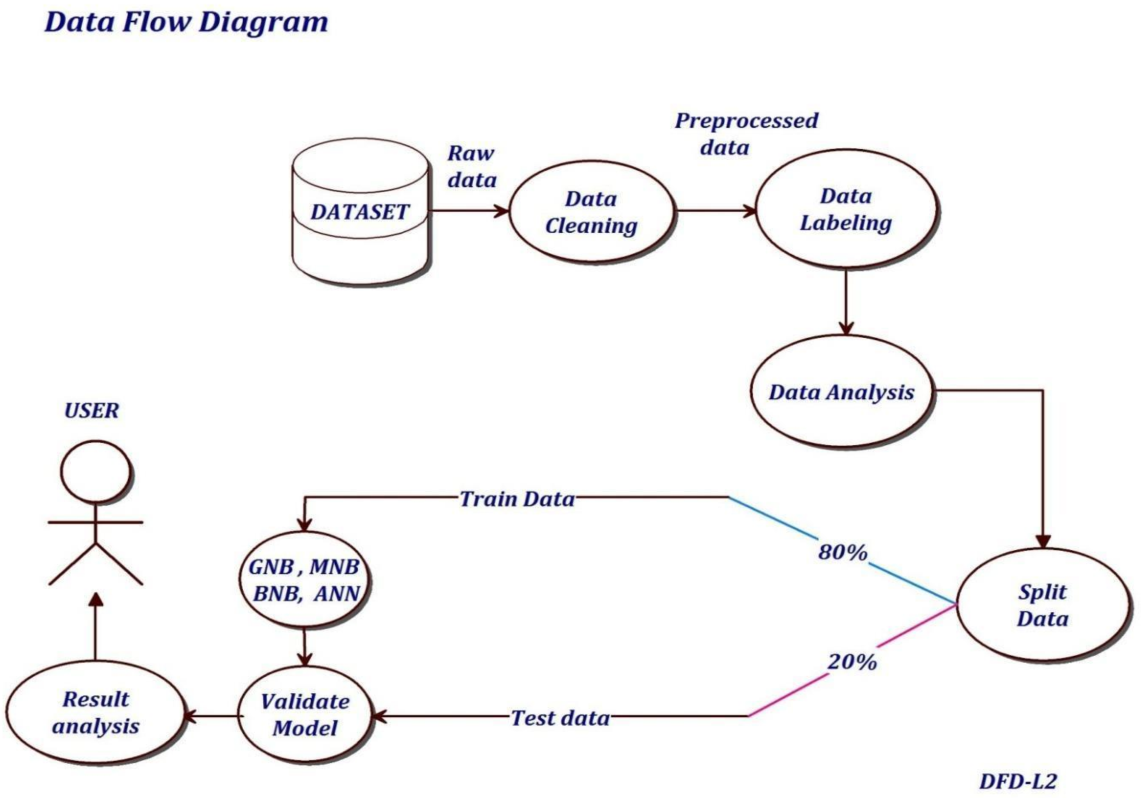


Fig 4.4.3 Level 2 Data Flow Diagram

Figure 4.4.3 shows a data flow diagram for pre-processing data. Raw data goes through a cleaning and labeling process to become preprocessed data ready for further analysis.

4.5 Sequence Diagram

A sequence diagram shows how objects interact in a specific order - that is, the sequence in which these interactions occur. It can also be referred to as event diagrams or event scenarios. These diagrams explain the functioning of objects in a system and the order in which they operate.

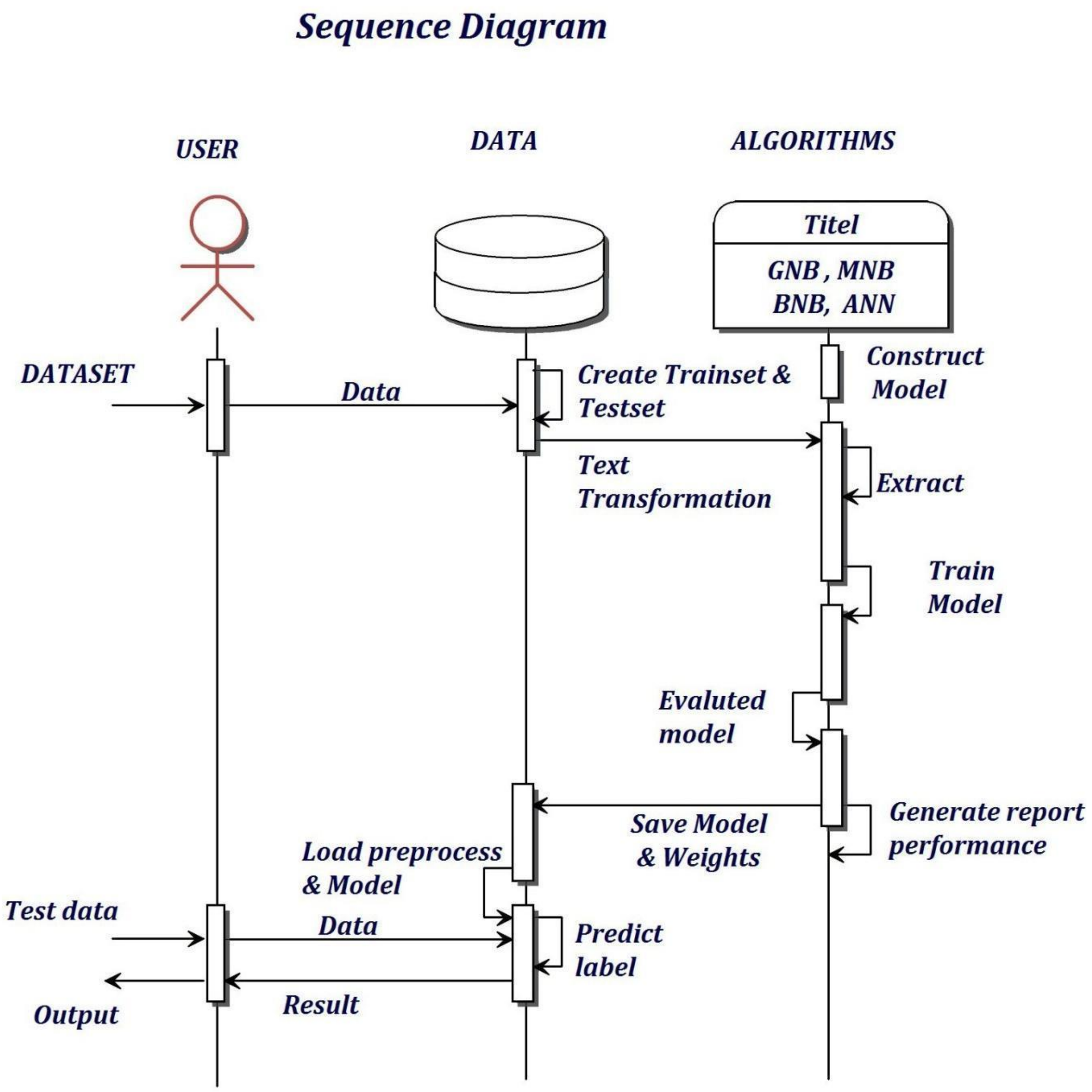


Figure 4.5. Sequence diagram

Figure 4.5 shows a sequence diagram illustrating the process of training the machine learning algorithm. The algorithm which is trained on a specific dataset and subsequently utilized to develop a model. This model is then assessed and stored for future reference. Following this, text information is employed to make predictions and produce a comprehensive report.

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1. Functional requirements

The functional requirements of a software system define the tasks it must perform and how it should behave when given certain inputs or conditions, such as calculations, data processing, and specific functions. These requirements are crucial for the overall functionality of the system. Once our model is verified, it should be able to accurately forecast the prices of seasonal items.

5.2. Non-functional requirements

Nonfunctional requirements outline the behavior and limitations of a system, called as the system quality for attributes. These attributes, including the performance, security, usability, and the compatibility, which are essential characteristics rather than specific features. They are inherent properties that result from the overall design and cannot be addressed by individual lines of code. Customer-requested attributes are detailed in the specification, and only relevant requirements should be included in our project.

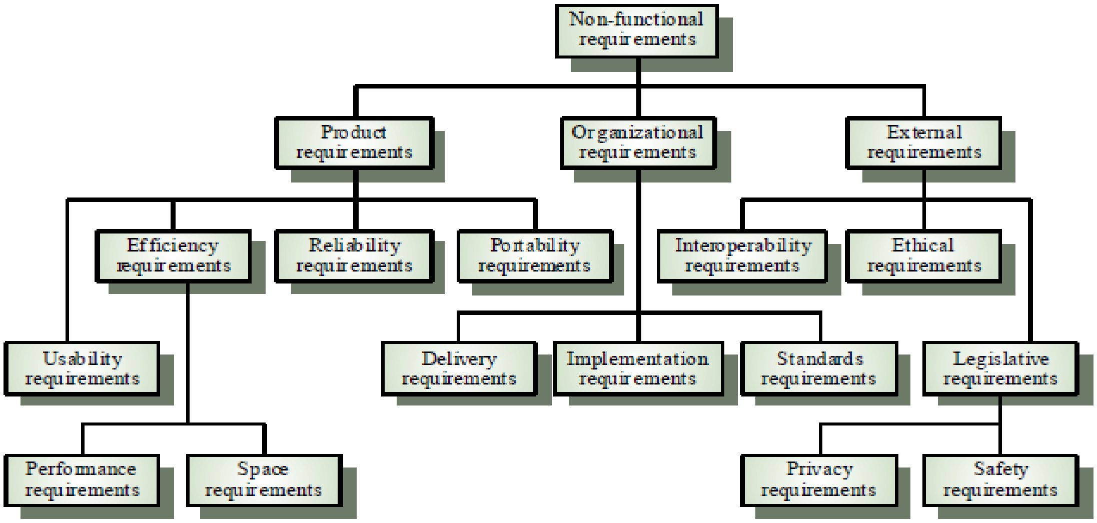


Figure 5.2. Non-functional requirements

Figure 5.2 The image shows a hierarchy of requirements for a product. It categorizes requirements into three main types: product, organizational, and external. Each main type branches out into more specific requirements, such as usability, performance, and safety.

Some Non Functional requirements are as follows:

Reliability

It is important for the structure to be strong and dependable in providing necessary functionalities. The changes made by the Programmer should be clearly visible in the structure when a customer has requested modifications. Both the Project leader and Test designer must ensure that the movements are implemented effectively.

Maintainability

The system maintenance and monitoring need to be a top priority. Having too many tasks running on different machines can make it difficult to ensure they are running smoothly and without any interruptions.

Performance

Many users will use the framework simultaneously. This poses a concern for performance as the system will run on one web server with a single database server in the background. The framework must be able to handle multiple users without any issues and provide fast access for all its users. For example, if there are two testers who are simultaneously trying to report a bug, then there should be no inconsistencies at the same time.

Portability

A framework which needs to be able to easily transition to another server if the current one encounters issues. This is necessary in case the web server hosting the framework experiences difficulties and the framework needs to be moved to a different server.

Scalability

The framework which needs to be flexible enough to add new features in the future. It should have the standard filter that can accommodate the new features.

Flexibility

Flexibility is like being able to roll with the punches and adjust to whatever life throws at you. An adaptable framework is one that can easily be changed or modified to suit different needs. By separating the different parts of a system, it makes it easier to make changes without affecting the entire system.

The Software Requirements specifications (SRS) is essentially which an organization's grasp of what the customer or potential where the client needs for their system before any design or the development starts. The data collected during analysis is turned into a document outlining the system's requirements. This document gives a concise overview of the system's desired services and operational constraints. In simpler terms, the SRS outlines what the software is supposed to do, not how it will do it. This insurance policy ensures that both the client and the organization are on the same page regarding their requirements at a specific time. The SRS document outlines the functions and constraints of a software system in clear language.

It acts as a guide for completing a project efficiently and cost-effectively. The Software requirements specification (SRS) which is considered the cornerstone document for all project management materials. It encompasses design specifications, work statements, software architecture plans, testing and validation strategies, and documentation plans.

The requirement which is something the system must meet. Requirement Management involves the gathering, organizing, and documenting the system's requirements clearly.

It is very important to recognize that the challenges in the requirements gathering may not always be the obvious and can come from the different sources.

Hardware Requirements

●	System processor	:	Core i3 / i5
●	Hard disk	:	500 GB
●	Ram	:	4 GB

✓ *Any desktop / Laptop system with the above configuration or higher level.*

Software Requirements

●	Operating System	:	Windows 10
●	Coding Langauage	:	Python
●	Tools	:	Anaconda
●	IDE	:	Jupyter Notebook

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

Methodology 1: Gaussian Naïve bayes

The Gaussian Naive Bayes algorithm is used for classifying data based on the probability of events happening according to Bayes' theorem and assuming a Gaussian distribution for numerical features. In predicting telecom customer churn, this algorithm which can help determine the likelihood of a customer leaving based on the factors such as usage habits, demographics, and interactions with services.

Methods: Popular tools such as Python's scikit-learn and data preprocessing libraries like pandas, matplotlib, seaborn, and NumPy are often used to effectively analyze telco churn patterns.

Methodology 2: Multinomial Naive Bayes

The multinomial naive bayes algorithm which is a different version of the Naive Bayes algorithm, made specifically for dealing with discrete data like word counts in text classification tasks. This method is ideal for scenarios in telecom customer churn where features are shown as counts or frequencies, for example, the amount of calls made, messages sent, or data used.

Methods: Python's scikit-learn and data preprocessing libraries such as pandas, matplotlib, seaborn, and NumPy are often utilized for analyzing telco churn patterns, offering valuable insights into the data.

Methodology 3: Bernoulli Naïve Bayes

Bernoulli Naive Bayes which is a version of the Naive Bayes algorithm, designed for binary-valued features commonly found in text classification tasks. It is utilized in telecom customer churn prediction when the features are binary indicators, like whether a customer has used a particular service or not.

Methods: Python's scikit-learn, along with data preprocessing libraries such as pandas, matplotlib, seaborn, and NumPy, are frequently used to analyze telco churn patterns and gain valuable insights.

Methodology 4: Artificial Neural Network(ANN)

Artificial Neural Networks (ANNs) which are famous for the being able to understand intricate connections within data. They excel at recognizing non-linear trends and are especially useful for analyzing extensive and varied datasets. For predicting customer churn in the telecom industry, an ANN is created to analyze multiple customer-related factors (like usage habits, demographics, and service interactions) to predict if a customer is probable to churn or stay.

Methods: Tools such as Python's scikit-learn and data preprocessing libraries like pandas, matplotlib, seaborn, NumPy, TensorFlow are frequently utilized for analyzing telco churn patterns, offering valuable insights.

Methodology 5: SMOTE (Synthetic Minority Oversampling Technique)

When we talk about machine learning, "smote" stands for Synthetic Minority Over-sampling Technique. This method helps address class imbalance by creating artificial instances of the minority class, improving the training of the model. SMOTE boosts the size of the dataset and enhances the model's accuracy, especially when working with imbalanced datasets, resulting in a more accurate representation of samples from the minority class.

CHAPTER7

EXPERIMENTATION

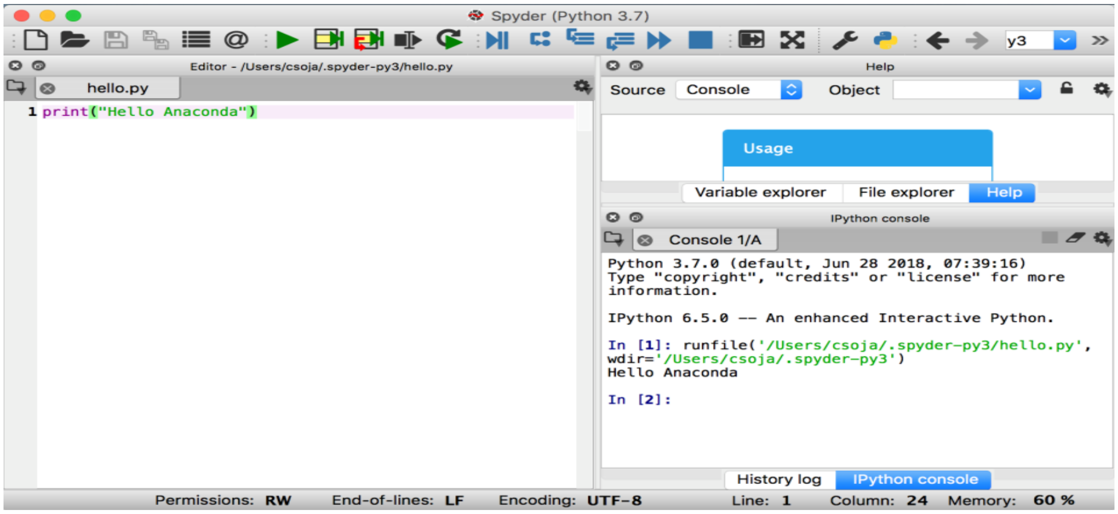
CHAPTER7 EXPERIMENTATION

Anaconda

The Anaconda Distribution which includes the conda and the Anaconda Navigator, along with the Python and numerous scientific packages. When you installed the Anaconda, you also installed these components. You have the option to experiment with both conda and Navigator to determine which best suits your needs for managing packages and environments. Additionally, you can easily switch between the two, and any tasks that conducted with one can also be accessed through the other.

try this the simple programming exercise, with the Navigator and the command line, to help you to decide which approach is the right for you.

Your first Python program: Hello Anaconda!



```
print("Hello Anaconda")
```

```
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
Type "copyright", "credits" or "license" for more
information.

IPython 6.5.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/csoja/.spyder-py3/hello.py',
wdir='/Users/csoja/.spyder-py3')
Hello Anaconda

In [2]:
```

Permissions: RW End-of-lines: LF Encoding: UTF-8 Line: 1 Column: 24 Memory: 60 %

Figure 7.1. Anaconda interface

Figure 7.1 shows Spyder, an integrated development environment (IDE) for Python. The code window displays a simple Python program that prints "Hello Anaconda." The console window below shows the output, "Hello Anaconda." In [2]: indicates the user is ready to enter new code. Text at the bottom right corner displays Python version and memory usage.

Close Spyder

From the Spyder's the top menu bar, we need to select Spyder - Quit Spyder (In macOS, select Python - Quit Spyder).

Run the Python in the Jupyter Notebook

1. On the Navigator's Home tab, in the Application pane on the right, scroll to Jupyter Notebook the tile and click the Install button to install the Jupyter Notebook

Jupyter Notebook (I D E)

The Jupyter Notebook which is an **interactive computing environment** that has enables the users to author notebook documents that include: - Live code - Interactive widgets - Plots - Narrative text - Equations - Images – Video.

These documents which provide a complete and self-contained record for a computation where that can be converted to the various formats and can be shared with others using the email, Dropbox, version control systems (like git/GitHub) or nbviewer.jupyter.org.

Components

The Jupyter Notebook which combines the three components:

The notebook web application: The interactive web applications for writing and running code interactively and authoring the notebook documents.

Kernels: Separate the processes started by the jupyter notebook the web application that runs users' code in the given language and also which returns the output back to the notebook web application. The kernel also handles things like computations for the interactive widgets, tab completion and introspection.

Notebook documents: Self-contained documents which that contain a representation of all content visible in the jupyter notebook web application, including inputs and outputs of the computations, narrative text, equations, images, and rich media representations of objects. Each notebook document has its own kernel.

Latex syntax in Markdown, which are rendered in the browser by MathJax.

Artificial intelligence

Artificial Intelligence which covers the wide range of technologies and the concepts. For example, some people might even classify the Dijkstra's shortest path algorithm as a form of AI. However, there is often confusion between two main categories within the AI: Machine Learning and Deep Learning. Both of these which involves using the statistical models to analyze data and make predictions. This article will explain why these two approaches are distinct and will clarify your understanding of these data analysis methods.

Machine learning

Machine Learning is the statistical learning approach that involves describing each of instances in the dataset with the set of features or the attributes. On the other hand, Deep Learning is the method that extracts the features or attributes from the raw data. Deep Learning utilizes the neural networks with multiple hidden layers, large amounts of the data, and powerful computational of the resources to achieve this. While the terms may seem similar, Deep Learning automatically which constructs representations of the data, whereas in Machine Learning, data representations are manually encoded as a set of features, requiring additional steps like the feature selection and extraction (e.g. PCA).Both terms discussed here are quite different from another group of traditional AI algorithms called Rule-Based Systems. In Rule-Based Systems, decisions are manually programmed to mimic a statistical model.

When it comes to Machine Learning and Deep Learning, you'll find various models that can be broadly categorized as the supervised and unsupervised. Unsupervised learning algorithms such as K-means, hierarchical clustering, and Gaussian mixture models are used to identify patterns and structures within data. On the other hand, supervised learning entails having labeled data where each instance in the dataset is associated with an output label, which can be either categorical or real-valued.When it comes to AI, regression models predict continuous values, while classification models predict distinct values. Binary classification models, for example, only have two possible outcomes - 1 for the positive and 0 for the negative. Some common supervised learning algorithms in the Machine Learning which includes linear regression, logistic regression, decision trees, support vector machines, neural networks, and the k-nearest neighbors.

Supervised Machine Learning

Most of the machine learning applications rely on supervised learning, which involves having the input variables (x) and an output variable (Y) and using an algorithm to learn how the input is mapped to the output.

The ultimate aim is to accurately approximate this mapping so that when the new input data (x) is received, the output variables (Y) can be predicted for that data.

Supervised learning is aptly named because it resembles a teacher guiding a student through the learning process. The algorithm learns from the training dataset by making predictions and receiving corrections from the teacher. The learning process continues iteratively until the algorithm reaches a satisfactory the level of performance.

Get the FREE Algorithms Mind Map

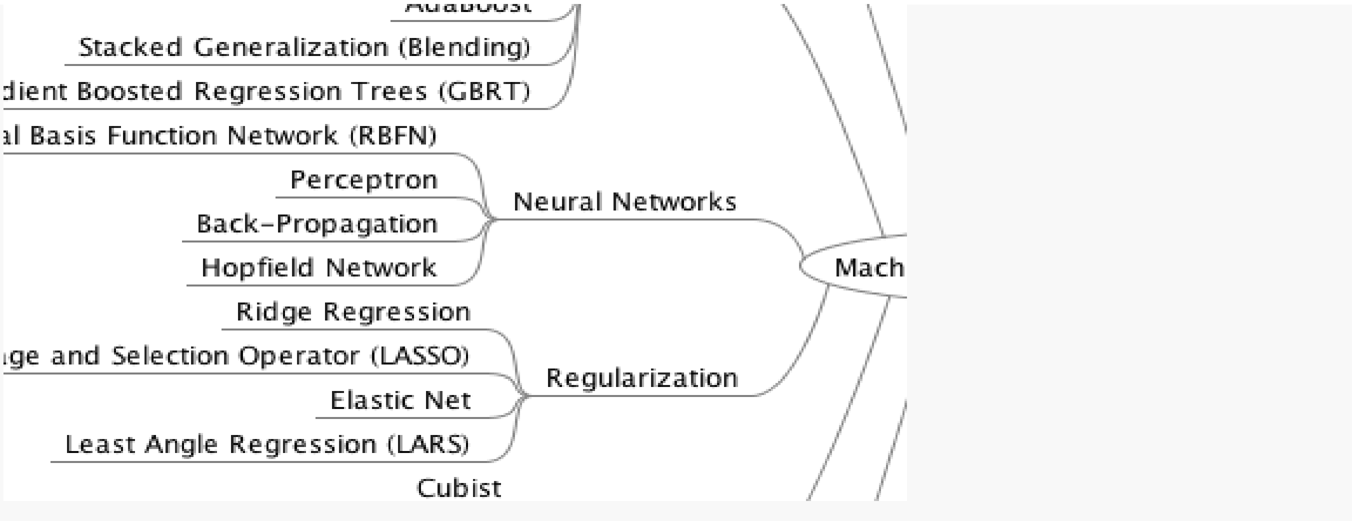

Stacked Generalization (Blending)
dient Boosted Regression Trees (GBRT)
al Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Neural Networks
Mach
Ridge Regression
ge and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Regularization
Cubist

Figure 7.2. the sample of the handy machine learning algorithms mind map.

Figure 7.2 shows a mind map of machine learning algorithms. It shows various algorithms like linear regression, neural networks, and decision trees.

Supervised learning problems that can be further grouped into regression and classification problems.

Classification: In machine learning, the classification problem which occurs when the output variables falls into the specific category, like "red" or "blue" or "disease" and "no disease". On the other hand, in the regression problem, the output variable is a numerical value, like "dollars" or "weight". Recommendation systems and time series prediction are some examples of problems that are based on classification and regression methods.

Here are a few common supervised machine learning algorithms: 1. Linear regression - used for solving regression problems. 2. Random forest - beneficial for both classification and regression tasks. 3. Support vector machines - great for handling classification problems.

Unsupervised Machine Learning

The Unsupervised learning algorithm which involves only input data (X) without the corresponding output variables. The objective of the unsupervised learning is to understand the underlying structure or distribution within the data to gain insights. In contrast to supervised learning, there are no correct answers or guidance provided to algorithms in unsupervised learning. Algorithms are tasked with exploring and revealing the inherent patterns in the data independently. Unsupervised learning tasks can be categorized as clustering and association problems.

Clustering: Have you ever wanted to find the natural groupings within data, like grouping customers based on their purchasing habits? That's what a clustering problem is all about.

Association: When we talk about association rule learning, we are looking at finding patterns in data that show relationships between different items. For example, if someone purchases item X, they are also likely to purchase item Y.

Here are some popular examples of the unsupervised learning algorithms are:

K-Means clustering problems.

The Apriori algorithm for association rule learning problems.

Semi-Supervised Machine Learning

Situations where there are a lot of data available (X) but only the portion of it is labeled (Y) are referred to as semi-supervised learning challenges. These types of problems lie midway between the supervised and unsupervised learning. Take, for instance, a collection of photos where only which certains images are labeled (e.g. dog, cat, person,rat,etc) while most remain unlabeled to those knowledgeable in the field. The acquisition and storage of unlabeled data is generally more cost-effective and straightforward. Unsupervised learning methods can be applied to identify and understand the patterns within the input data.

One way to make predictions for data without labels is by using supervised learning techniques. This involves making educated guesses, incorporating the new data back into the algorithm as training data, and using the model to predict outcomes for future unseen data.

Data Size:

The Machine Learning and the Deep Learning both excel at processing large datasets, but Machine Learning is more suitable for smaller datasets. For instance, when dealing with the just 100 data points, algorithms like decision trees and k-nearest neighbors are more effective than training a deep neural network. This is because of the issue of Interpretability.

Interpretability:

Here is the example for how the interpretability works in ML & DL:

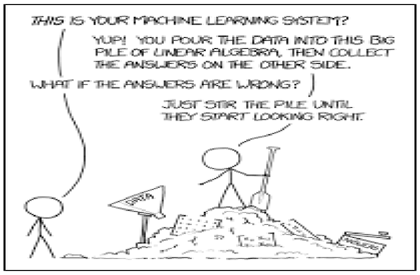

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Figure 7.3. Interpretability

Figure 7.3 The cartoon depicts a man standing on a pile of dirt shoveling data into it. Text above the pile reads "This is your machine learning system?" Text below reads "Just shovel the pile until they start looking right." The cartoon implies machine learning involves large amounts of data.

Many people find it hard to understand deep learning methods and machine learning algorithms like Support Vector Machine or Naive Bayes because they are difficult to interpret. For instance, in a dog vs. cat problem, if a Convolutional Neural Network outputs 'cat', it is unclear why it made that decision. On the other hand, when using machine learning techniques to analyze data like an electronic health record or bank loan dataset, it is easier to grasp the rationale behind the model's predictions.

Decision trees are a great example of interpretability in machine learning. You can follow logical tests down nodes of the tree to make a decision. Another highly interpretable algorithm is k-Nearest Neighbors, which is non-parametric but still considered a machine learning algorithm. It is easy to reason about similar instances with this algorithm.

SDLC

This document will provide a thorough explanation of the methodology being utilized to successfully complete and optimize this project. Various methodologies and discoveries in this field are typically documented in journals to benefit others and enhance future research. The methodology employed is aimed at achieving the project's objectives and delivering exceptional results. The evaluation of this project will be based on the System Development Life Cycle (SDLC), which typically involves three key steps: planning, implementation, and analysis.

Software Development Life Cycle (SDLC) is a method used by software organizations to build and maintain software. It includes a detailed plan for developing, maintaining, replacing, and improving specific software. The life cycle outlines a process for enhancing software quality and overall development.

Software Development Life Cycle (SDLC) outlines the activities to be carried out at different points by a software professional. It guarantees that the final result will meet the client's needs and remain within the financial constraints. Therefore, having a good understanding of this development process is crucial for software professionals.

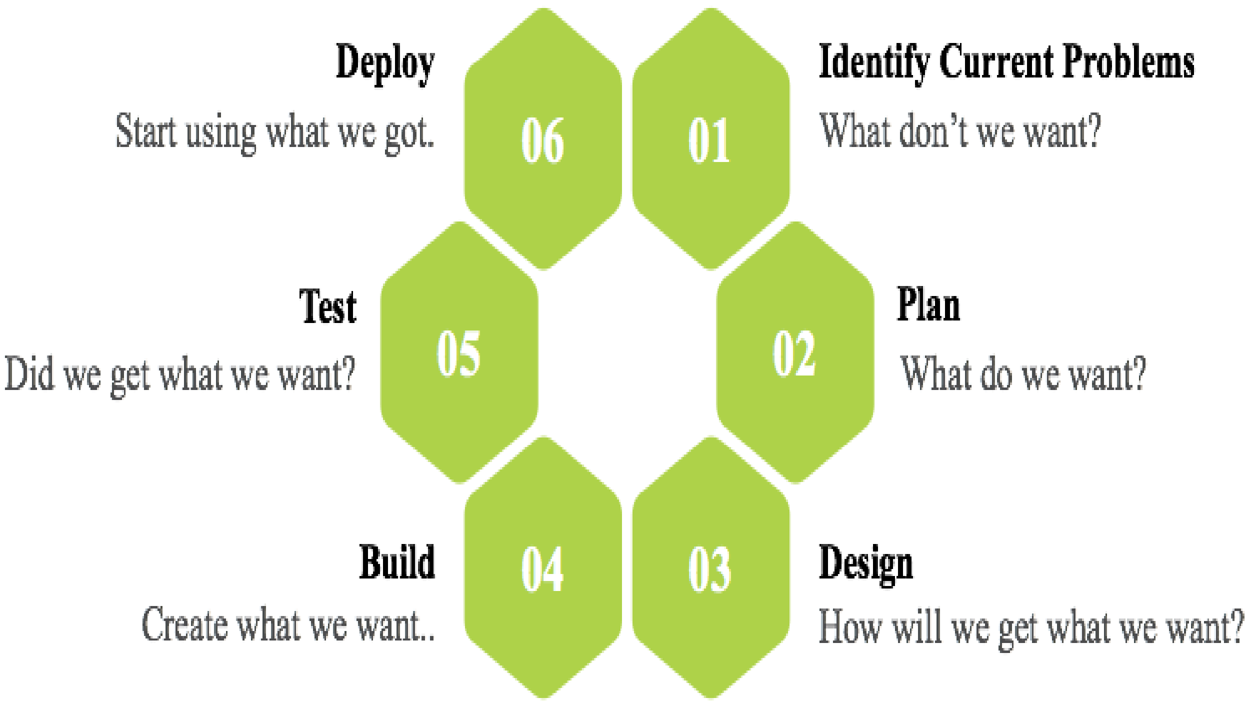


Fig 7.4. Software Development Life Cycle

Figure 7.4 The image shows a circular diagram with six green hexagons representing steps to achieve a goal. It starts with identifying what you want and ends with deploying it.

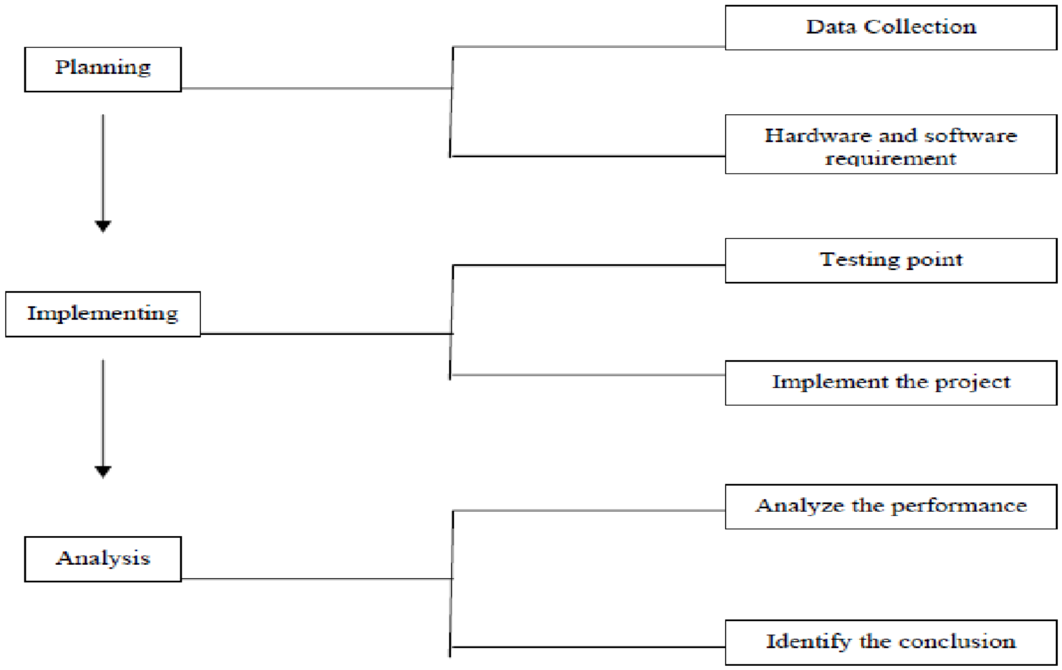


Fig 7.5. Steps of Methodology

Figure 7.5 shows a flowchart outlining the stages of planning, implementing, and analyzing a project. It highlights defining requirements, testing, and identifying conclusions.

Planning:

In order to address all necessary information and needs for hardware and software, proper planning is essential. The planning process consists of two key components: data collection and determining hardware and software requirements.

Data collection:

In order for the machine learning to be successful, it requires two main components: data and the models. It is a crucial to gather a large amount of data with a variety of features that can be used to train the learning model effectively. The more data you have, the more accurate your predictions will be, so it is important to ensure that you have enough data points to work with.

The data that is collected from the online sources is initially in its raw form, consisting of the statements, numbers, and qualitative terms. This raw data which often contains errors, omissions, and inconsistencies that needs to be addressed through careful analysis of the completed questionnaires. The processing of the primary data involves several steps, including grouping a large volume of raw data from field surveys based on similar details of individual responses.

Data preprocessing is the method which is used to clean the raw data before analysis. When data is collected from various sources, it is in a raw format that is not suitable for analysis. The steps are taken to convert the data into a tidy dataset before iterative analysis begins. This process, known as data preprocessing, includes -

Data Cleaning

Data Integration

Data Transformation

Data Reduction

The Data Preprocessing is necessary steps because of the presence of the unformatted real-world data. Mostly real-world data is composed of -

Inaccurate data (Missing data) - There are many reasons for the missing data such as data which is not continuously collected, a mistake in data entry, technical problems with biometrics and much more.

The presence of the noisy data (erroneous data and outliers) - The reasons for the existence of the noisy data could be a technological problem of a gadget that gathers the data, a human mistake during data entry and much more.

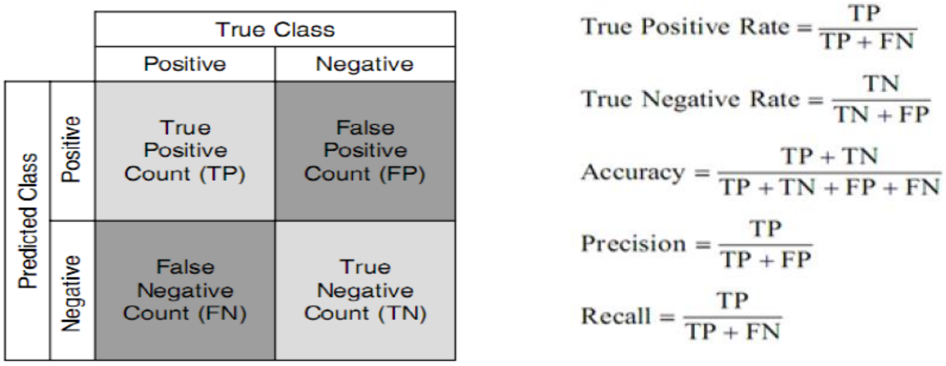
Inconsistent data - There are inconsistencies present in the data for various reasons, including duplicates, errors in data entry by humans, mistakes in codes or names, and violations of data constraints, among others.

Analysis

In this last stage, we will assess our classification model using the image dataset we have prepared and analyze its performance. To measure the effectiveness of our classification, we will use accuracy as a metric to compare it with existing methods.

After creating a model, understanding how well it can predict on new data is crucial. Once a predictive model is trained on past data, one naturally wants to know how it will fare on unseen data. It is common to experiment with different types of models for the same prediction task and compare their performance (e.g., accuracy) to choose the best one for real-world decision making. Performance metrics like accuracy and recall are commonly used to evaluate a predictor. Let's first discuss these common performance metrics before delving into some popular models.

"Performance Metrics for Predictive Modeling In classification tasks, the main way to measure performance is through a confusion matrix (also known as a classification matrix or contingency table). The following diagram illustrates a confusion matrix for a binary classification problem. It also provides formulas for the most commonly used metrics that can be derived from the confusion matrix"



		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

$$\text{True Positive Rate} = \frac{TP}{TP + FN}$$

$$\text{True Negative Rate} = \frac{TN}{TN + FP}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Figure 7.6. confusion matrix and formulae

Figure 7.6 shows a confusion matrix used in machine learning. It compares predicted classifications to actual classifications for positive and negative classes. This helps evaluate a model's performance.

As seen in the above figure, In classification models, the diagonal numbers indicate correct decisions while the numbers outside the diagonal represent errors. The true positive rate, also known as hit rate or recall, is calculated by dividing the correctly identified positives by the total positive instances. The false positive rate, also known as false alarm rate, is calculated by dividing the incorrectly identified negatives by the total negative instances. Overall accuracy is calculated by dividing the total correctly identified instances by the total number of samples.

The structure of a Convolutional Neural Network (ConvNet) is similar to how neurons are connected in the human brain, taking inspiration from the way the Visual Cortex is organized. Each neuron in the network only responds to stimuli from a specific area of the visual field, called the Receptive Field. These fields come together to map out the entire visual space.

Flexibility

There are times when you might prefer to create your own custom elements rather than relying on what already exists. For instance, you may want to define your own cost function, metric, or layer. While Keras 2 offers extensive flexibility for implementation, it is widely recognized that low-level libraries offer even greater customization options. The same holds true for TensorFlow, allowing for more extensive tweaking compared to Keras.

Functionality

While Keras offers basic functionalities for constructing deep learning models, TensorFlow surpasses it by providing more advanced operations. This is particularly useful for researchers and developers working on specialized deep learning models. Here are a few examples of high-level operations included in TensorFlow:

Threading and Queues

Queues are a great tool for processing tensors separately in a graph. You can also run multiple threads within a Session for parallel computations, improving the efficiency of your operations.

Debugger

TensorFlow has an additional feature that sets it apart: a dedicated debugger that offers a unique view into the inner workings of TensorFlow graphs as they run. This tool is invaluable for pinpointing and resolving a wide range of bugs that may arise during training or inference.

Control

Having control over your network allows you to have a deeper insight into its functioning. TensorFlow provides the tools necessary to maintain this level of control. With TF, you have the ability to manipulate different aspects of your network with ease, such as adjusting weights or gradients.

Numpy

NumPy, short for Numerical Python, is a powerful library that includes multidimensional array objects and a variety of tools for working with these arrays. With NumPy, you can easily perform mathematical and logical operations on arrays.

This tutorial will cover the basics of NumPy, including its structure and capabilities, as well as various array functions and indexing methods. We will also provide an introduction to Matplotlib, a plotting library used in conjunction with NumPy. Throughout the tutorial, we will use examples to help you grasp the concepts more easily. In simpler terms, NumPy is a Python package that provides tools for working with numerical data. It is essential for tasks like data analysis, scientific computing, and machine learning.

Numeric, NumPy, the predecessor of Numeric, was created by Jim Hugunin. Another package, Numarray, was developed with added functionality. In 2005, Travis Oliphant merged the features of Numarray into the Numeric package to create NumPy. This open source project has had numerous contributors.

When developers use NumPy, they can carry out a variety of tasks such as performing mathematical and logical operations on arrays, conducting Fourier transforms, manipulating shapes, and working with linear algebra operations. NumPy also provides built-in functions for linear algebra and generating random numbers.

MatLab is commonly used for technical computations, but Python has become popular as a more modern and versatile programming language. Python's NumPy, which serves as an alternative to MatLab, is advantageous due to being open source.

In NumPy, the main building block is the N-dimensional array, often called ndarray. This array holds a collection of elements that all have the same type, and can be accessed using an index starting from zero. Each element in the ndarray takes up a fixed amount of memory space, with each element being of a particular data type (dtype). When you extract an element from an ndarray using slicing, it is presented as a Python object of one of the scalar types specific to arrays. The connection between ndarrays and data types is illustrated in the diagram below, highlighting the relationship between the object (dtype) and array scalar type.

Matplotlib

Matplotlib is a great tool for creating visualizations in Python, particularly for 2D plots of arrays. It is a versatile data visualization library that works well with NumPy arrays and fits smoothly into the larger SciPy ecosystem. Developed by John Hunter in 2002, Matplotlib offers users the ability to easily interpret large datasets through visually appealing graphics. The library includes various types of plots such as line charts, bar graphs, scatter plots, histograms, and more.

Installation:

Matplotlib and its dependencies are available as wheel packages for Windows, Linux, and macOS distributions. To install the matplotlib package, execute the following command:

```
python -mpip install -U matplotlib
```

Import a matplotlib:

```
from matplotlib import pyplot as plt
or
import matplotlib.pyplot as plt
```

Plots in Matplotlib:

Matplotlib provides a wide array of plots that are helpful for understanding quantitative data by identifying trends, patterns, and relationships. These visualizations help make sense of numerical data and offer valuable insights into different phenomena.

Applications of Matplotlib

For those who are familiar with using programs, Matplotlib offers an interface similar to MATLAB, especially when paired with IPython. This allows for detailed control over things like line styles, fonts, axes, and more, either through an object-oriented approach or by using functions that are similar to those found in MATLAB. Matplotlib is a Python library designed for creating high-quality 2D plots that are suitable for professional use. It can be used for both interactive and non-interactive plotting and allows for images to be saved in formats like PNG and PS. It can be seamlessly integrated with various window toolkits like GTK+, wxWidgets, and Qt, and offers a wide variety of plot types including lines, bars, pie charts, histograms, and more.

Matplotlib is well-known for its flexibility, ability to be customized, and user-friendly interface. It can be easily integrated into various types of scripts, applications, or web pages. Additionally, it enables interactive usage with the Python interpreter or IPython.

Advantages of Matplotlib

Matplotlib was conceptualized with a straightforward goal: simplifying the straightforward and enabling the accomplishment of complex tasks, as articulated by its creator and leader, John Hunter. Originally influenced by established scientific computing tools like gnuplot and MATLAB, Matplotlib sought to emulate the functionalities of MATLAB, a prevalent graphing tool in the field. This similarity attracted numerous users, enticing them to transition from MATLAB to Matplotlib.

It uses Python: Python is captivating for scientific pursuits due to its interpreted nature, high-level syntax, user-friendly learning curve, extensive customization capabilities, and comprehensive standard library. Prominent entities such as NASA, JPL, Google, DreamWorks, and Disney have embraced Python for their endeavors.

The software is free to use, so there are no licensing fees involved, which is great for educators and students on tight budgets. Unlike MATLAB, Python is a fully-fledged programming language with more versatile functionalities. Python offers a wide range of additional modules that can assist in various data tasks, making it a comprehensive tool for data acquisition, analysis, and visualization. Matplotlib, a Python library, is highly customizable and adaptable to different needs, offering a variety of graph types, features, and settings.

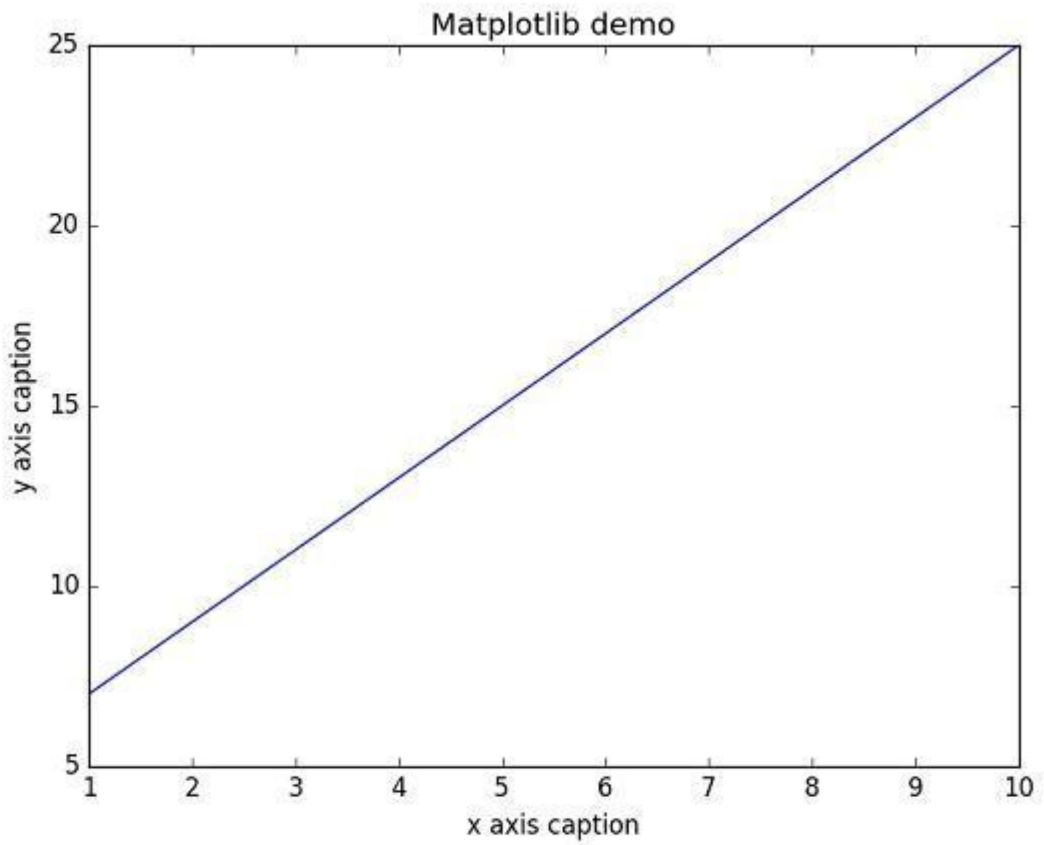


Figure 7.7. Matplotlib demo

Figure 7.7 shows a line graph. The x-axis caption is missing, but the y-axis. The line shows a curve that increases steeply at first, then flattens out around 20.

To show values discreetly in a plot, use a format string with the plot() function instead of a straight line graph. NumPy's numpy.histogram() function offers a visual of data frequency distribution, showing equal horizontal rectangles for class intervals (bins) and varying heights for frequencies. The numpy.histogram() function needs an input array and bins as arguments, with the bin array defining the boundaries for each bin.

```
import numpy as np

data = np.array([22, 87, 5, 43, 56, 73, 55, 54, 11, 20, 51, 5, 79, 31, 27])

histogram, bins = np.histogram(data, bins=[0, 20, 40, 60, 80, 100])
```

```
print(histogram)

print(bins)
```

The output of the above code is−

```
[3 4 5 2 1]
[ 0 20 40 60 80 100]
```

If you have numerical data and want to create a histogram graph, you can use Matplotlib's 'plt()' function from the pyplot submodule. Simply provide the function with an array of data and a bin array, and it will generate a histogram for you.

S K Learn

Introduction

Scikit-learn is a Python library with classes and functions that can be imported into Python programs. Proficiency in Python programming is necessary to use scikit-learn well. However, it does not have a command-line interface or graphical user interface for non-programmers. Scikit-learn is well-known as the top Python library for machine learning, utilizing NumPy, SciPy, and matplotlib for efficient tools in machine learning and statistical modeling. These tools cover classification, regression, clustering, and dimensionality reduction tasks.

Scikit-learn provides a range of supervised and unsupervised learning algorithms through a user-friendly Python interface. It is available on different Linux distributions with a simplified BSD license, making it popular in academic and commercial fields. Known as a Python library for data analysis and mining, scikit-learn is open-source and licensed under BSD. It uses machine learning libraries like NumPy to handle multi-dimensional arrays and matrices, helping users with data exploration and modeling tasks.

Core API

Scikit-learn has a simple and consistent API that includes three main interfaces: an estimator for creating and training models, a predictor for making predictions, and a transformer for converting data. In this article, we will explore these interfaces after discussing our basic principles and data representation techniques. Our design philosophy focuses on making the code framework easy to understand, using clear conventions, and minimizing the required methods for each object.

Consistency: All items, whether simple or complex, have a unified interface consisting of a small number of methods. This interface is consistently documented for all items.

Inspection: Scikit-learn's learning algorithms use constructor parameters and parameter values to make decisions. These values are saved and can be easily accessed as public attributes.

Non-proliferation of classes: In scikit-learn, learning algorithms are depicted through custom classes, while datasets are typically represented as NumPy arrays or SciPy sparse matrices. Hyperparameter names and values are expressed as standard Python strings or numbers, ensuring scikit-learn remains user-friendly and interoperable with other libraries.

Composition: Many tasks in machine learning can be represented as a series of data transformations or combinations. Some algorithms are essentially meta-algorithms that rely on other algorithms for their parameters. These algorithms are usually built using pre-existing components whenever possible, with sensible defaults in place. If a user-defined parameter is needed for an operation, The library automatically assigns a starting value to ensure the task is completed in a clear and straightforward way, offering a simple solution for the job at hand.

Components of scikit-learn:

Scikit-learn offers many features to help you understand data distribution.

Estimators: These are algorithms that can be used for building models or fitting to data. They include classifiers, regressors, clustering algorithms, and transformers.

Transformers: Transformers are used for data preprocessing, feature engineering, feature selection, and dimensionality reduction.

Datasets: Scikit-learn provides a collection of datasets for practicing and learning machine learning algorithms.

Model Evaluation: The library offers tools for evaluating model performance through metrics like accuracy, precision, recall, F1-score, and more.

Model Selection: Scikit-learn provides utilities for selecting the best models and tuning hyperparameters through techniques like cross-validation and grid search.

Utilities: The library contains different tools for tasks like pre-processing, extracting features, and managing imbalanced data.

Supervised Learning algorithm: Scikit-learn offers a wide range of supervised learning algorithms, such as Linear Regression, SVM, Decision Trees, and Bayesian approaches. The variety of algorithms available is a major factor in the popularity of scikit-learn. I personally began using scikit-learn for supervised learning tasks and I strongly suggest it to individuals new to scikit and machine learning.

Cross-validation: There are multiple ways to measure how accurate supervised models are when applied to new data. Cross-validation is a commonly used method to evaluate the accuracy of supervised models on unseen data.

Unsupervised learning algorithm: There are plenty of algorithms to choose from, ranging from the clustering and factor analysis to principal component analysis and the unsupervised neural networks.

Various toy datasets: We found using datasets like IRIS and Boston House prices dataset really helpful when We was learning scikit-learn and had previously mastered SAS through academic projects. It made learning new libraries much easier.

Feature extraction: Extracting features is important in capturing key elements from images and text, using methods like Bag of Words.

Estimators

The estimator interface is a key component of the library, guiding the creation of objects and offering a fit method for training models with data. It covers all types of learning algorithms, including supervised and unsupervised ones like classification, regression, and clustering. Additionally, the library offers estimators for tasks like feature extraction, feature selection, and dimensionality reduction.

Predictor

It is common to prepare or structure data before using it in a machine learning model. In the library, there are many functions called transformers that have a transform function. This function takes in new data called X_test and creates a changed version of it. The library provides various transformers that are designed for tasks such as preparing, selecting features, extracting features, and reducing dimensionality.

Transformers

Before inputting data into a learning algorithm, it is common practice to preprocess or arrange the data. The scikit-learn library offers various functions that adhere to a transformer format, including a transform function. This function takes new data, X_test, and produces an altered version of X_test. The library includes a range of transformers that are meant for tasks such as preprocessing, selecting features, extracting features, and reducing dimensionality.

Advanced API

The scikit-learn API provides advanced features for creating meta-estimators, building complex estimators, and choosing models. Meta-estimators.

"Many machine learning algorithms are essentially meta-algorithms that rely on simpler algorithms as parameters. For instance, ensemble methods involve creating and merging multiple simple models (such as decision trees), while multiclass and multi-label classification techniques can extend a binary classifier to handle multiple classes or labels."

Pipelines and feature unions

The scikit-learn API stands out for its ability to create new estimators by combining multiple base estimators, which allows for integrating common machine learning processes into a single entity. This entity functions as an estimator and can be used in the same way as traditional estimators.

Extending scikit-learn

The scikit-learn API stands out for its ability to generate novel estimators through the combination of multiple base estimators. This compositional approach enables the consolidation of various machine learning procedures into a unified entity, serving as an estimator with the same usability as standard ones.

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

8.1. Test Cases

Table 8.1. Testcases

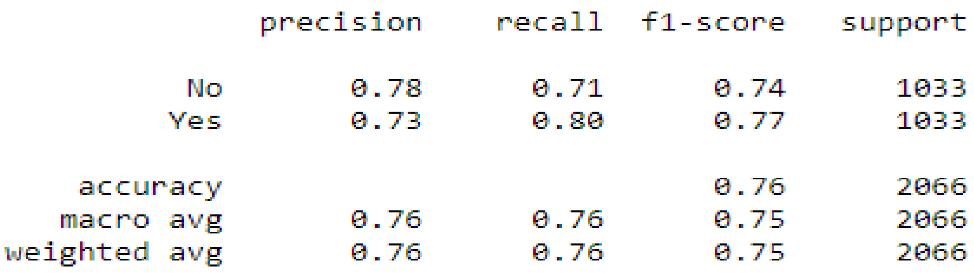
Id	**Test Case Title**	**Test Input**	**Result**	**Remark s**
TC_1	A Data Upload Datasets File Path	File Uploaded	Successfully	Pass
TC _2	Data Cleaning	Raw Dataset	Cleaned the Data	Pass
TC _3	Data Preparation for the Training	Dataset and Split Ratio	Train-set and the Test-set created successfully	Pass
TC _4	Model construction and Training	Training Algorithm And Train-set	Model trained successfully using the train-set	Pass
TC _5	Model Validation	Trained the Model And Test-set	Display model validation parameters with its values	Pass
TC _6	Display the Results	Model Performance Statistics	Classification accuracy and error rate with plot	Pass

Table 8.1 shows a table summarizing test case execution results. It lists test case IDs, titles, inputs, results (pass or fail), and remarks.

8.2. Results

8.2.1. Guassian Naive Bayes

Guassian Naive Bayes(GNN) is commonly used in predicting customer churn. It calculates the probability of a customer leaving by analyzing factors such as usage behaviors and demographics. This algorithm operates under the assumption that these factors are unrelated and follow a normal distribution. While it is effective for analyzing vast amounts of data, it may struggle with closely related features.



```
              precision    recall  f1-score   support

          No       0.78      0.71      0.74      1033
         Yes       0.73      0.80      0.77      1033

    accuracy                           0.76      2066
   macro avg       0.76      0.76      0.75      2066
weighted avg       0.76      0.76      0.75      2066
```

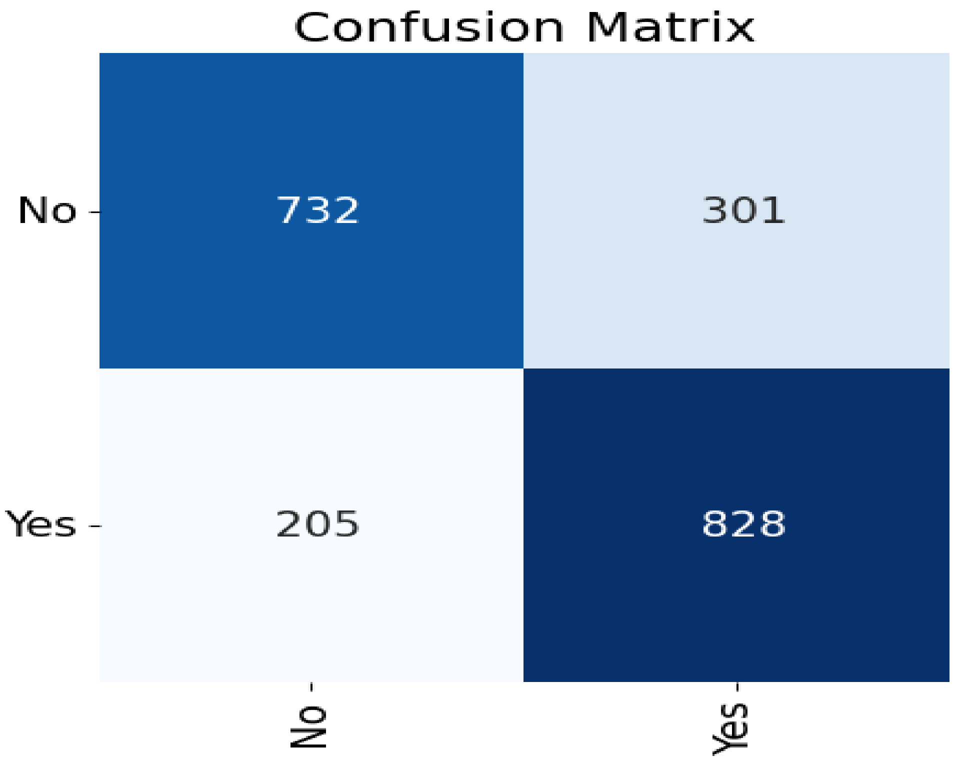
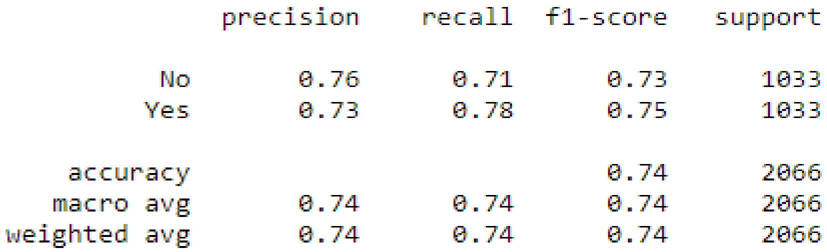


Figure 8.2.1. Classification report and confusion matrix for Gaussian naïve bayes

Figure 8.2.1 shows a confusion matrix. It shows the number of people classified as saying "Yes" or "No" when the actual answer was "Yes" or "No". For example, 732 people were predicted to say "No" and actually said "No"

8.2.2. Multinomial Naive Bayes

Multinomial Naive Bayes is a version of the Naive Bayes model that is tailored for dealing with discrete data, like word frequencies in text classification tasks. In the context of telecom customer churn, this method is best used when features are measured as counts or frequencies, such as call volumes, messages sent, or data consumption.



```
              precision    recall  f1-score   support

          No       0.76      0.71      0.73      1033
         Yes       0.73      0.78      0.75      1033

    accuracy                           0.74      2066
   macro avg       0.74      0.74      0.74      2066
weighted avg       0.74      0.74      0.74      2066
```

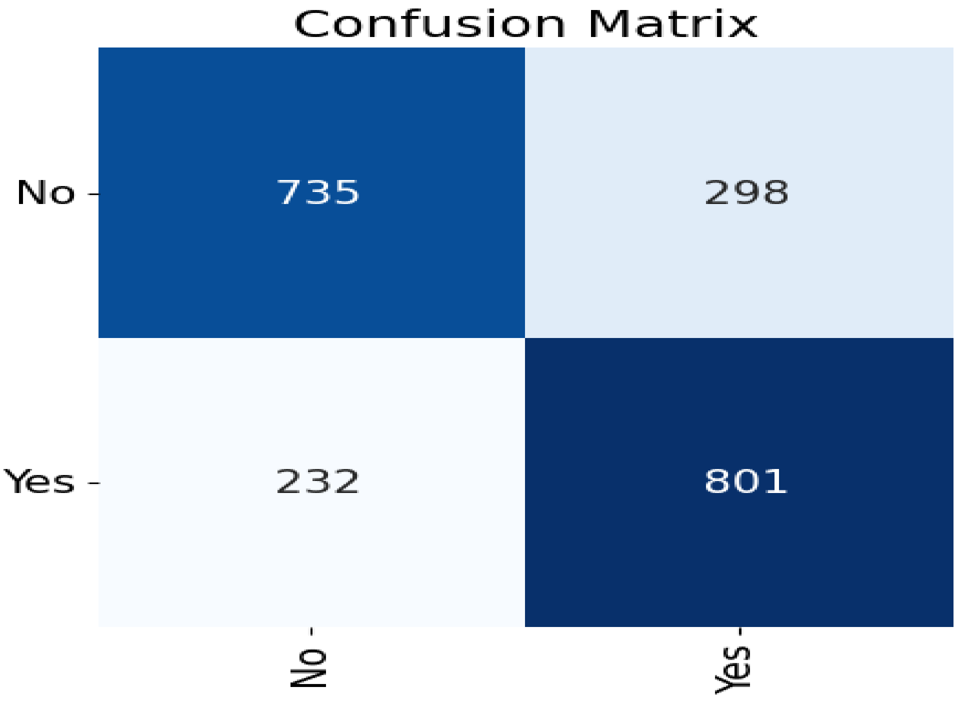
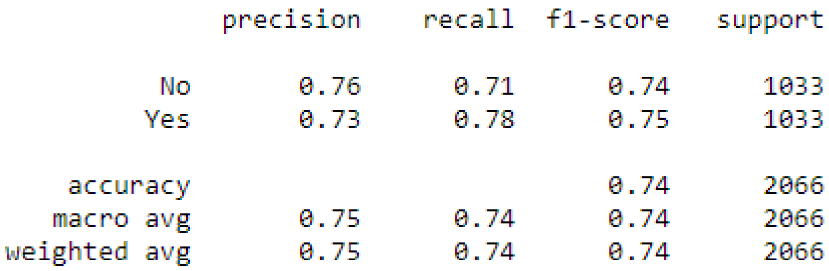


Figure 8.2.2. Classification report and confusion matrix for Multinomial naïve bayes

Figure 8.2.2 shows a confusion matrix. It shows the performance of a binary classification model. The rows represent the actual classes ("Yes" or "No") and the columns represent the predicted classes. The confusion matrix shows high accuracy (735 and 801) and relatively low errors (298 and 232).

8.2.3. Bernoulli naïve bayes

Bernoulli Naive Bayes is a type of algorithm similar to the Naive Bayes algorithm, designed for binary features and commonly used in text classification tasks. In the context of predicting customer churn in the telecom industry, this algorithm is used when features are binary, such as whether a customer has used a particular service or not. This model is helpful for datasets with mostly binary features and can accurately predict customer churn based on binary behaviors or service usage patterns.



```
              precision    recall  f1-score   support

          No       0.76      0.71      0.74      1033
         Yes       0.73      0.78      0.75      1033

    accuracy                           0.74      2066
   macro avg       0.75      0.74      0.74      2066
weighted avg       0.75      0.74      0.74      2066
```

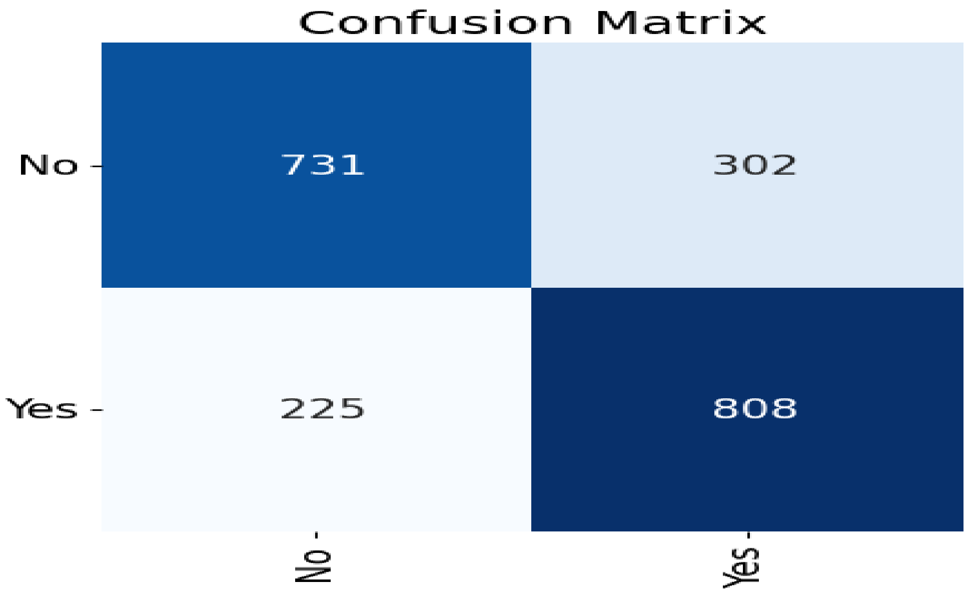
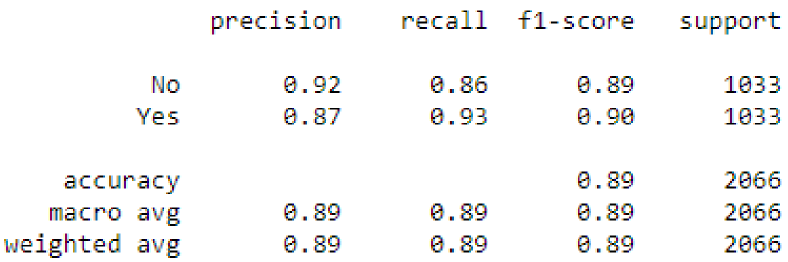


Figure 8.2.3.. Classification report and confusion matrix for Bernoulli naïve bayes

Figure 8.2.3 The image is a confusion matrix, likely for an m-BERT model evaluating fake news detection in Dravidian languages. It shows high accuracy in classifying real and fake news ("No" and "Yes") with most values on the diagonal.

8.2.4. Artificial neural network

Artificial neural networks are famous for their knack at understanding intricate connections within data. They excel at deciphering non-linear patterns and shine when working with vast and varied datasets. In the realm of telecom customer churn prediction, an ANN is crafted to analyze multiple customer features (like usage habits, demographics, and service engagements) to predict if a customer will likely churn.



```
              precision    recall  f1-score   support

          No       0.92      0.86      0.89      1033
         Yes       0.87      0.93      0.90      1033

    accuracy                           0.89      2066
   macro avg       0.89      0.89      0.89      2066
weighted avg       0.89      0.89      0.89      2066
```

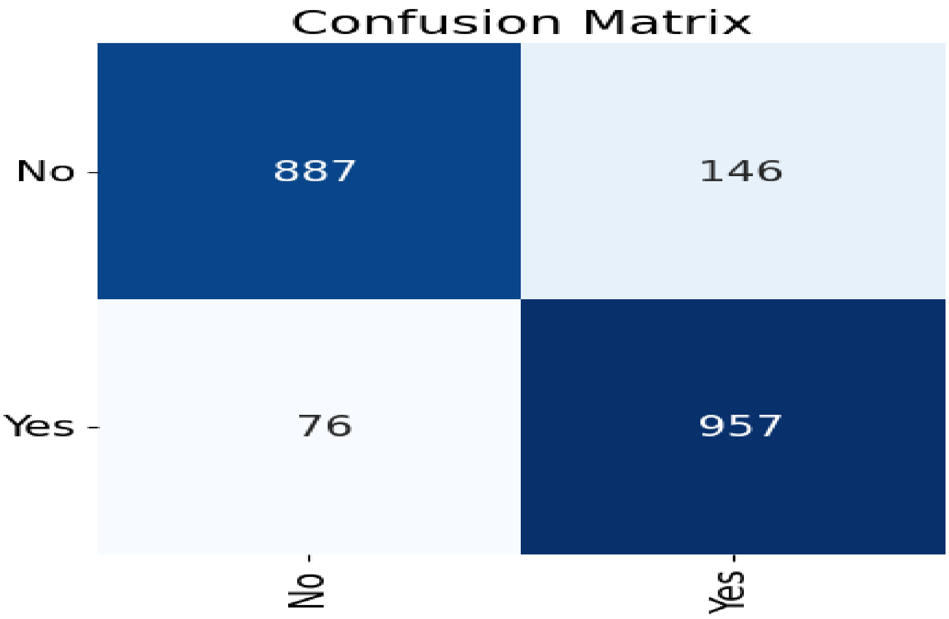


Figure 8.2.4. Classification report and confusion matrix for Artificial neural network

Figure 8.2.4 shows a confusion matrix, likely for an m-BERT model evaluating fake news detection in Dravidian languages. It shows high accuracy, with most values on the diagonal. For example, 887 real news articles ("No") were correctly classified and 957 fake news articles ("Yes") were correctly classified.

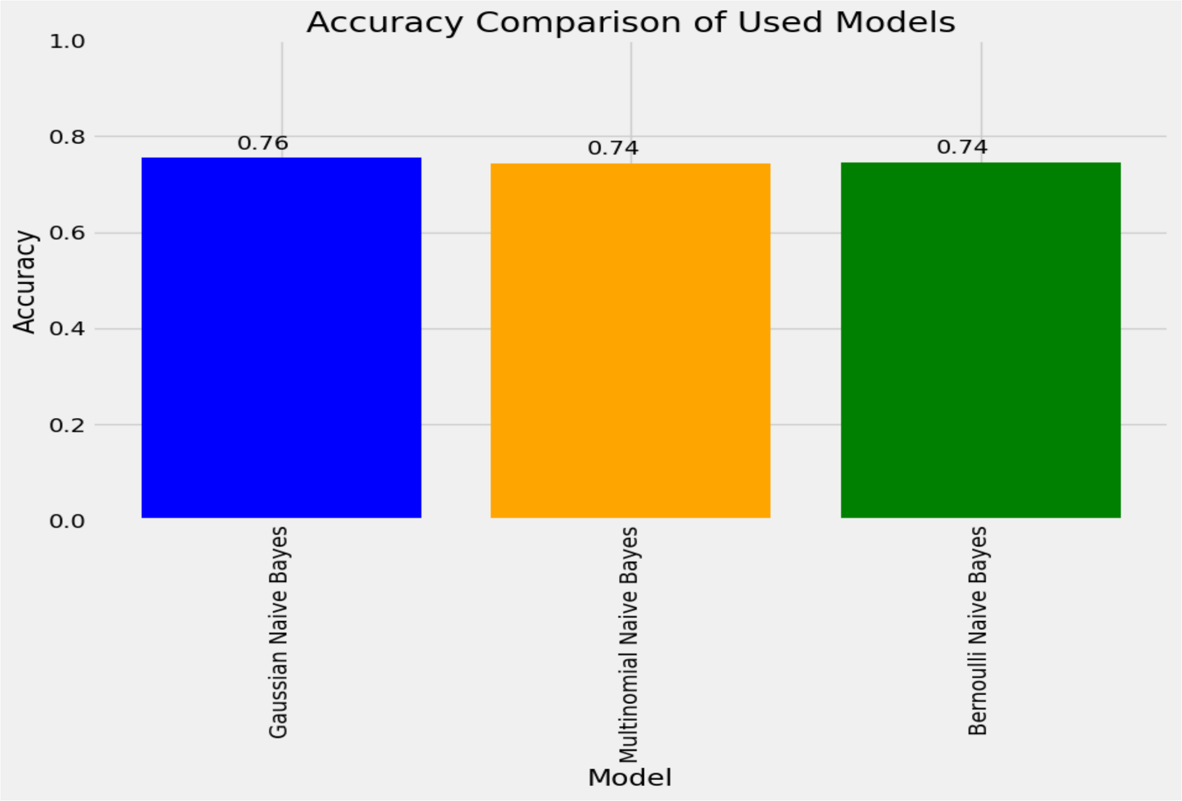


Figure 8.2.5. Accuracy comparison of use models

Figure 8.2.5 shows a bar graph comparing the accuracy of three different machine learning models. The green rectangle model has the highest accuracy at 0.76, followed by the blue rectangle model at 0.74, and the yellow rectangle model at 0.70.

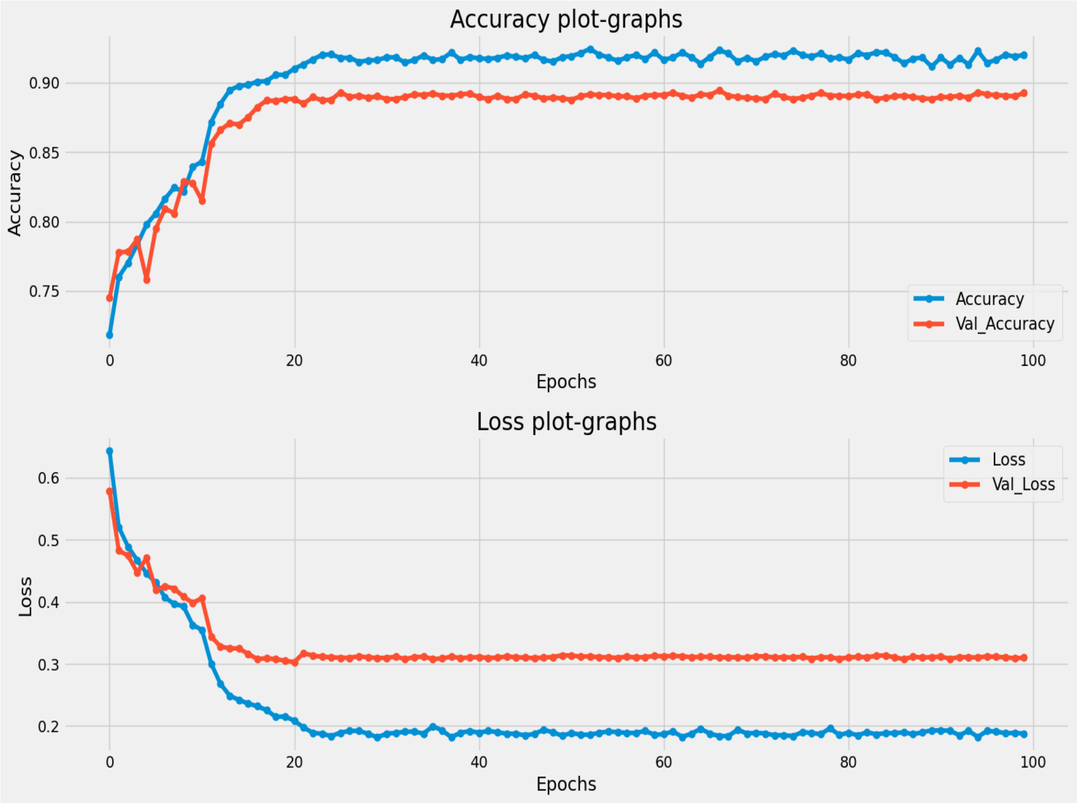


Figure 8.2.6. Accuracy plot graph

Figure 8.2.6 there is a graph that displays the training and validation accuracy and loss of a machine learning model. The accuracy curves, represented by blue and green, typically show an upward trend over time, while the loss curves, indicated by orange and red, generally show a decreasing trend. This indicates that the model is making progress and learning.

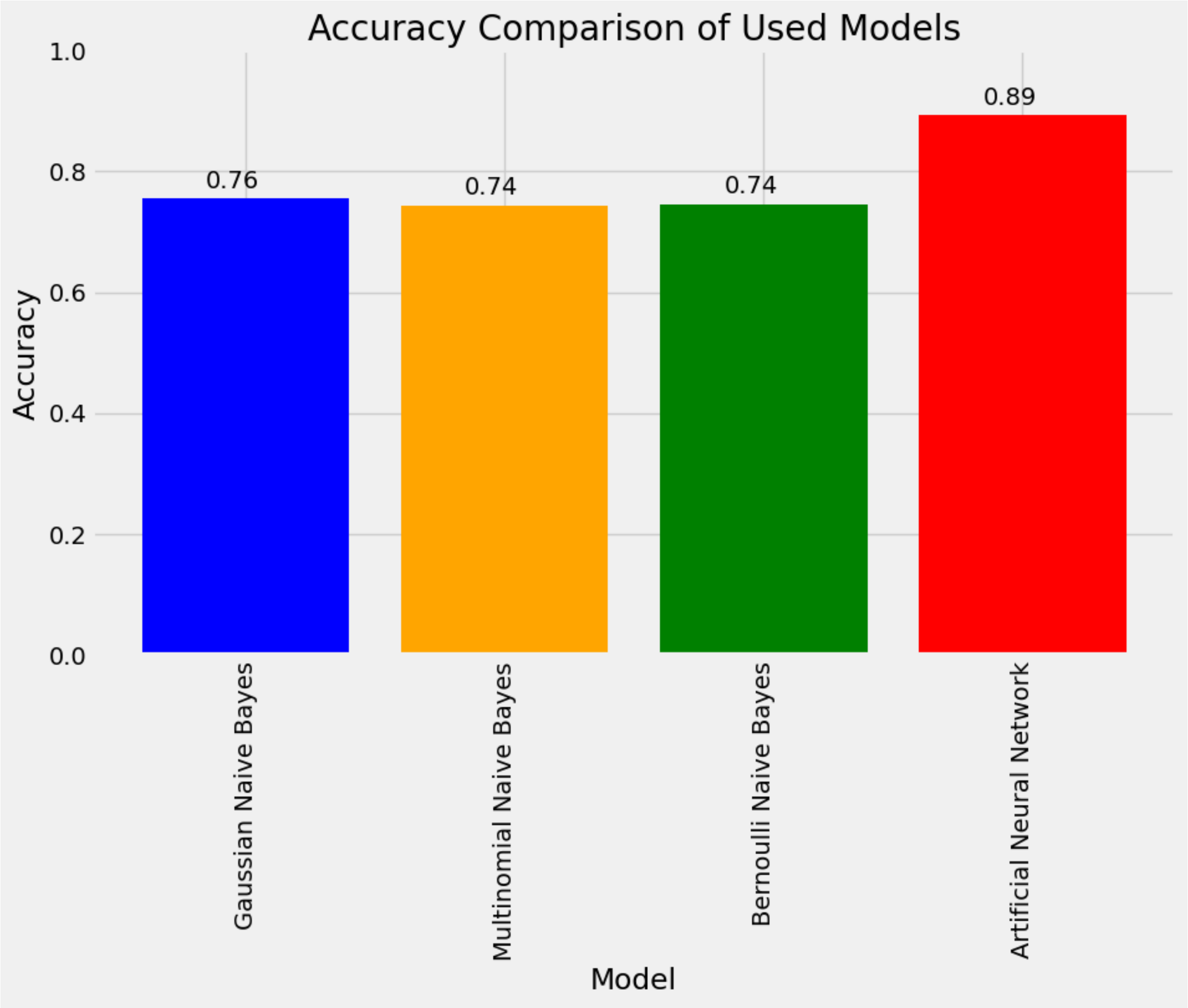


Figure 8.2.7. Accuracy comparison of use models

Figure 8.2.7 The bar graph shows the accuracy of four machine learning models. The red bar model, labeled "Artificial Neural Network," has the highest accuracy at 0.89. The other three models have lower accuracy ratings.

REFERENCES

[1] Chaithra K N, Manu M N, Shrikanth N G, Anupama K/ Minimization of Churn Rate Through Analysis of Machine Learning. International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), 29 April 2023

[2] SHULI WU 1, WEI-CHUEN YAU 1, (Member, IEEE), THIAN-SONG ONG 2 , (Senior Member, IEEE), AND SIEW-CHIN CHONG 2 , (Senior Member, IEEE) Integrated Churn Prediction and Customer Segmentation Framework for Telco Business April 29, 2021.

[3] J. Vijaya1 · E. Sivasankar An efficient system for customer churn prediction through particle swarm optimization-based feature selection model with simulated annealing. Springer Science+Business Media, LLC, 19 September 2017

[4] Georgina Esteves, Joao Mendes-Moreira Churn Prediction in the Telecom Business. The eleventh international conference on digital information management (ICDIM) 2016

[5] Awodele Oludele, Adeniyi Ben*, Ogbonna A.C., Kuyoro S.O., Ebiesuwa Seun Enhanced Churn Prediction in the Telecommunication Industry. International Journal of Innovative Research in Computer Science & Technology (IJIRCST), March 2020

[6] Ankita Zadoo, Tanmay Jagtap, Nikhil Khule, Ashutosh Kedari, Shilpa Khedkar A review on Churn Prediction and Customer Segmentation using Machine Learning. International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), 26-27 May 2022

[7] Ammar A. Q. Ahmed, Maheswari D. Churn prediction on huge telecom data using hybrid firefly-based classification. Egyptian Informatics Journal, 1 March 2017

[8] Authors: Adnan Amin, Sajid Anwar, Awais adnan, Muhammad Nawaz, newton Howard, junaid qadir, (senior member, ieee), Ahmad hawalah, and Amir Hussain, (senior member, ieee) Comparing Oversampling Techniques to Handle the Class Imbalance Problem.26 October, 2016

[9] Paula Branco, Luís Torgo, Rita P. Ribeiro A Survey of Predictive Modelling under Imbalanced Distributions. Association for Computing Machinery New York,13 August 2016

[10] Adnan Amin, Feras Al-Obeidat, Babar Shah, May Al Tae, Changez Khan, Hamood Ur Rehman Durrani, Sajid Anwar Just-in-time customer churn prediction in the telecommunication sector. The Journal of Supercomputing · June 2020

https://github.com/Komal102002/DSU-CSE-Major.git