"Customer Churn Prediction"

A project report submitted in partial fulfillment of the requirements

for the award of credits to

Python a Skill Enhancement Course of

Bachelor of Technology

In

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING (CSM)

By

JILLELLA SUSMITHA

23BQ1A4266

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING (CSM)

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(Approved by AICTE and permanently affiliated to JNTUK)

Accredited by NBA and NAAC with 'A' Grade

NAMBUR (V), PEDAKAKANI (M), GUNTUR-522 508

JUNE 2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING (CSM)

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY: NAMBUR

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

CERTIFICATE

This is to certify that the project titled "**Customer Churn Prediction**" is a bonafide record of work done by **Jillella Susmitha** under the guidance of **Ms.B.Lalitha Rajeswari, Assistant Professor** in partial fulfillment of the requirement for the award of credits to **Machine Learning** course of Bachelor of Technology in Computer Science & Engineering – Artificial Intelligence & Machine Learning (CSM), JNTUK during the academic year 2024-25.

B. Lalitha Rajeswari — **Course Instructor**

Prof. K. Suresh Babu — **Head of the Department**

DECLARATION

I, Jillella Susmitha (23BQ1A4266), hereby declare that the Project Report entitled **Customer Churn Prediction**" done by me under the guidance of Ms.B.Lalitha Rajeswari, **Assistant Professor** is submitted in partial fulfillment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING (CSM).**

DATE : **SIGNATURE OF THE CANDIDATE**

PLACE : **JILLELLA SUSMITHA**

ACKNOWLEDGEMENT

We express our sincere thanks wherever it is due

We express our sincere thanks to the Chairman, Vasireddy Venkatadri Institute of Technology, Sri Vasireddy Vidya Sagar for providing us well equipped infrastructure and environment.

We thank Dr. Y. Mallikarjuna Reddy, Principal, Vasireddy Venkatadri Institute of Technology, Nambur, for providing us the resources for carrying out the project.

We express our sincere thanks to Dr. K. Giribabu, Dean of Academics for providing support and stimulating environment for developing the project.

Our sincere thanks to Dr. K. Suresh Babu, Head of the Department, Department of CSM, for his co-operation and guidance which helped us to make our project successful and complete in all aspects.

We also express our sincere thanks and are grateful to our guide Mr. M. Pardha Saradhi, Associate Professor, Department of CSM, for motivating us to make our project successful and fully complete. We are grateful for his precious guidance and suggestions.

We also place our floral gratitude to all other teaching staff and lab technicians for their constant support and advice throughout the project.

NAME OF THE CANDIDATE
JILLELLA SUSMITHA
23BQ1A4266

TABLE OF CONTENTS

ABSTRACT

Customer churn prediction is an essential analytical task in the telecom sector, as retaining existing customers is significantly more cost-effective than acquiring new ones. This project presents a machine learning-based approach to predict customer churn using the Telco Customer Churn dataset. The project workflow includes data collection, exploratory data analysis (EDA), data cleaning, and feature engineering. Key preprocessing steps include the conversion of string-based numerical fields, handling missing values, and encoding categorical variables using Label Encoding. Since the dataset is imbalanced, the SMOTE technique is employed to balance the churn and non-churn classes. Various classification algorithms such as Decision Tree, Random Forest, and KNN Classifier are implemented to identify the best-performing model. These models are evaluated using standard classification metrics including accuracy, confusion matrix, and classification report (precision, recall, F1-score). The trained model is serialized using pickle, allowing for deployment and reuse in real-time systems. This end-to-end project demonstrates how machine learning can be effectively used to gain business insights and improve customer retention strategies in the telecom domain.

1. INTRODUCTION

1.1 Overview of Customer Churn

What is Customer Churn?

Customer churn, also known as customer attrition, refers to the phenomenon where customers stop using a company's products or services over a given period. In the telecommunications industry, churn typically means a subscriber cancels their service or switches to a competitor. This is a critical business metric because acquiring new customers generally costs significantly more than retaining existing ones. High churn rates can negatively impact a company's revenue, profitability, and market share.

Churn can occur for various reasons such as dissatisfaction with service quality, better offers from competitors, pricing issues, or changes in customer needs. Understanding and predicting churn behavior allows companies to take proactive measures like targeted marketing, personalized offers, or improved customer service to retain valuable customers and reduce revenue loss.

1.2 Importance of Churn Prediction

Churn prediction plays a crucial role in many industries, especially in telecommunications, finance, subscription services, and e-commerce. It involves identifying customers who are likely to stop using a company's products or services in the near future. The importance of churn prediction can be summarized as follows:

Revenue Retention

Acquiring new customers is often more expensive than retaining existing ones. By predicting churn, companies can proactively engage at-risk customers, reducing revenue loss.

Cost Efficiency

Targeted retention campaigns based on churn prediction allow companies to allocate marketing and customer service resources more efficiently, focusing efforts where they will have the highest impact.

Improved Customer Experience

Understanding why customers churn helps businesses improve products, services, and customer support, leading to higher satisfaction and loyalty.

Competitive Advantage

Companies that successfully predict and reduce churn maintain a stable customer base, which can improve market share and brand reputation compared to competitors.

Better Strategic Planning

Churn prediction provides insights into customer behavior and preferences, enabling companies to design better pricing, promotions, and product features.

Data-Driven Decision Making

Leveraging data and machine learning models for churn prediction helps organizations make informed decisions rather than relying on intuition.

1.3 Problem Statement

In the highly competitive telecom industry, customer retention is a major challenge, as losing customers can significantly impact revenue. Identifying customers who are likely to churn in advance allows telecom companies to implement targeted retention strategies. This project aims to develop a machine learning model that can accurately predict customer churn based on historical customer data from the Telco Customer Churn dataset. The goal is to use this predictive model to support decision-making processes that improve customer satisfaction and reduce churn rates.

Dataset Description

The dataset used for this project is the Telco Customer Churn Dataset, which contains detailed information about customers of a telecommunications company. It includes various attributes related to customer demographics, account details, service subscriptions, and usage behavior. The dataset provides a comprehensive view of customer profiles and their interactions with the company, making it suitable for building predictive models to identify churn patterns.

Key features in the dataset typically include customer ID, gender, age, tenure (length of time the customer has stayed with the company), types of services subscribed (such as phone, internet, and streaming services), payment methods, monthly charges, total charges, and

whether the customer has churned. This rich set of features enables the analysis of factors contributing to customer churn and supports the development of effective retention strategies.

1.4 Machine Learning Process in Loan Status Prediction

1. **Data Collection:**

   The Telco Customer Churn dataset is used, containing customer demographics, service details, contact information, and churn status. This comprehensive data enables analysis of customer behavior and identification of potential churn indicators. It provides the foundation for building predictive models.

2. **Data Preprocessing:**

   Preprocessing includes converting string-based numeric fields to appropriate data types, handling missing values to maintain data integrity, and encoding categorical variables using Label Encoding.

   Due to class imbalance, the SMOTE technique is applied to balance the number of churn and non-churn samples. These steps prepare the data for accurate model training.

3. **Exploratory Data Analysis(EDA):**

   EDA is conducted to understand feature distributions, spot patterns, and explore relationships between variables and churn. Visualizations and statistical summaries help uncover key indicators influencing customer churn. This step guides effective feature selection and model design.

4. **Model Selection:**

   Three classification algorithms—Decision Tree, Random Forest, and KNNClassifier—are implemented to evaluate their effectiveness in predicting churn. These models are chosen for their ability to handle complex feature relationships and provide interpretable outputs. Model comparison helps in selecting the best-performing algorithm.

5. **Training the Model:**

   The dataset is split into training and testing sets, and the models are trained on the processed data using supervised learning techniques. SMOTE-balanced data ensures both churn and non-churn classes are learned effectively. Training focuses on optimizing performance through parameter tuning

6. **Hyperparameter Tuning:**

   Use RandomizedSearchCV to perform exhaustive search over specified parameter values for each model, optimizing their performance through cross-validation.

7. **Model Evaluation:**
   Evaluate trained models on the testing set using metrics like accuracy, precision, recall, F1-score, and confusion matrix to assess predictive capability.
8. **Model Deployment (Optional):**
   Integrate the best-performing model into a real-world application to automate loan approval decisions.

1.5 Scope

This project focuses on developing a machine learning-based system to predict customer churn in the telecom sector using the Telco Customer Churn dataset. It covers the complete data science pipeline, including data preprocessing, exploratory data analysis, feature engineering, model building, evaluation, and deployment. The project is limited to classification models such as Decision Tree, Random Forest, and KNNClassifier, with SMOTE used to handle class imbalance. The model is designed to be integrated into real-time systems, enabling telecom companies to proactively identify customers at risk of churning and take timely retention measures. The scope also includes model serialization for future use but does not cover large-scale production deployment or advanced deep learning techniques.

1.6 Terminology Overview

Model:
Classification models like Decision Tree, Random Forest, and KNNClassifier were used to predict customer churn. These models were chosen for their efficiency, accuracy, and interpretability.

Feature:
Input features include customer demographics, service usage, contract type, and billing information. Categorical features were label encoded, and numerical fields were cleaned and standardized.

Target:
The target variable is **Churn**, indicating whether a customer left the service (Yes) or stayed (No). It's a binary classification problem handled using machine learning techniques.

Training:
The dataset was split into training and test sets, and SMOTE was applied to balance the classes. Models were trained using Python libraries like Scikit-learn and KNNClassifier.

Prediction:

The trained model predicts churn likelihood for new customer data. The final model is saved using Pickle for integration into real-time telecom systems.

2. SOFTWARE REQUIREMENTS

To implement the Customer Churn prediction system using machine learning techniques, the following software components are required. These tools provide the necessary environment for data preprocessing, model training, evaluation, and visualization.

2.1 Programming Language: Python

Python is the core programming language used in this project due to its simplicity, flexibility, and powerful ecosystem for data science and machine learning. It supports a wide range of libraries such as Pandas and NumPy for data handling, Scikit-learn for implementing machine learning models, and Matplotlib and Seaborn for data visualization. Python's compatibility with tools like Jupyter Notebook also enables interactive coding and easier debugging, making it highly suitable for end-to-end churn prediction workflows.

2.2 Development Environment: Jupyter Notebook

Jupyter Notebook is the development environment used in this project, providing an interactive and user-friendly interface for writing and executing Python code. It allows seamless integration of code, visualizations, and narrative text within a single document, making it ideal for data analysis and machine learning workflows. With features like inline plotting, step-by-step execution, and easy data inspection, Jupyter Notebook enhances productivity during exploratory data analysis, model development, and result interpretation. Its support for real-time output and markdown annotations also makes it effective for documenting the churn prediction process and sharing insights.

2.3 Required Python Libraries

This project uses a range of Python libraries essential for building an effective machine learning pipeline. NumPy and Pandas handle numerical operations and data manipulation. Matplotlib and Seaborn support data visualization during exploratory analysis. Scikit-learn provides tools for preprocessing, model training, and evaluation. Additional libraries like Imbalanced-learn and KNNClassifier enhance model performance and handle class imbalance.

Library/Package	Purpose
NumPy	Performs numerical computations and supports multi-dimensional array operations.
Pandas	Used for data manipulation, cleaning, and handling structured/tabular datasets.
Matplotlib	Creates static, animated, and interactive visualizations for EDA.
Seaborn	Enhances Matplotlib by providing attractive and easy-to-use statistical plots.
Scikit-learn	Provides tools for data preprocessing, classification models, evaluation metrics, and hyperparameter tuning (e.g., RandomizedSearchCV).
Imbalanced-learn	Used for handling class imbalance through techniques like SMOTE (Synthetic Minority Over-sampling Technique).

2.3.1 NumPy

NumPy (short for *Numerical Python*) is a core library in the Python ecosystem for scientific and numerical computing. It provides highly efficient tools to perform operations on large multi-dimensional arrays and matrices. NumPy is designed for performance and supports vectorized operations, which means tasks can be performed much faster compared to traditional Python loops. It is widely used in data science, machine learning, image processing, and many scientific applications due to its flexibility and speed.

In the context of machine learning projects like customer churn prediction, NumPy is often used during data preprocessing, mathematical transformations, and intermediate computations.

Key Features of NumPy:

- Provides n-dimensional array objects (ndarray) for storing homogeneous data.
- Supports vectorized operations for faster numerical computations without explicit loops.

- Includes built-in mathematical functions for operations like mean, standard deviation, dot product, etc.
- Offers functionality for linear algebra, Fourier transforms, and random number generation.
- Integrates seamlessly with libraries like Pandas, Scikit-learn, and Matplotlib.

Use of NumPy in This Project:

- Efficient handling of numerical features from the dataset.
- Applied for intermediate computations in model training and evaluation.

2.3.2 Pandas Library

Pandas is a powerful and widely-used Python library for data manipulation, analysis, and preparation. It provides flexible data structures such as Series (1D) and DataFrame (2D) that allow efficient handling and analysis of structured or tabular data. Pandas is especially useful in data preprocessing stages, including cleaning, transforming, filtering, and summarizing datasets, which are essential steps in any machine learning project.

In the context of customer churn prediction, Pandas is used extensively for loading the dataset, exploring the data through statistics and visualizations, handling missing values, encoding categorical features, and preparing the final dataset for machine learning algorithms.

Key Features of Pandas:

- Provides DataFrame and Series data structures for labeled data handling.
- Easy data filtering, sorting, grouping, and aggregation functions.
- Supports reading and writing data from various file formats like CSV, Excel, and SQL.
- Built-in methods for handling missing data, data type conversions, and data merging.
- Integrates seamlessly with other libraries like NumPy, Matplotlib, and Scikit-learn.

Use of Pandas in This Project:

- Loaded the Telco Customer Churn dataset into a DataFrame.
- Cleaned and preprocessed the data, including handling missing values.
- Performed **EDA** (Exploratory Data Analysis) using summary statistics and filtering.

- Converted categorical columns into numerical format using encoding techniques.
- Prepared the final dataset for training machine learning models.

2.3.3 Matplotlib

Matplotlib – a fundamental Python library used for data visualization and Exploratory Data Analysis (EDA). It helps create a wide variety of plots and charts, such as line graphs, bar charts, histograms, and scatter plots, enabling users to visually explore data patterns, trends, and distributions effectively. Matplotlib's flexibility and customization options make it essential for interpreting and presenting data insights clearly.

- Visualize the distribution of customer demographics like age, gender, and region.
- Plot histograms and bar charts to explore features such as tenure, monthly charges, and contract types.
- Compare churn rates across different customer groups using pie charts or count plots.
- Identify patterns and correlations between variables that impact churn.
- Detect outliers or anomalies in data through scatter plots or box plots.

2.3.4 seaborn

Seaborn is a Python data visualization library built on top of Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics with less code. Seaborn simplifies complex visualization tasks by offering built-in themes, color palettes, and functions tailored for visualizing relationships between variables, distributions, and categorical data. It's especially useful for Exploratory Data Analysis (EDA) and helps uncover patterns and insights in data more easily.

2.3.5 Scikit-learn

Scikit-learn is a powerful and widely used open-source machine learning library in Python. It provides simple and efficient tools for data mining, data analysis, and machine learning, making it ideal for building predictive models.

In your customer churn prediction project, Scikit-learn is used for key tasks such as:

- Preprocessing: Handling missing values, encoding categorical variables, splitting datasets, and scaling features.

- Modeling: Implementing machine learning algorithms like Decision Tree and Random Forest.
- Evaluation: Generating performance metrics such as accuracy, confusion matrix, precision, recall, and F1-score.

It integrates well with libraries like NumPy and Pandas, making it highly suitable for end-to-end machine learning workflows.

Imbalanced-learn – SMOTE

Imbalanced-learn is a Python library specifically designed to handle imbalanced datasets in classification problems. One of its most commonly used techniques is **SMOTE (Synthetic Minority Over-sampling Technique)**, which generates synthetic examples for the minority class instead of simply duplicating existing ones. This helps balance the dataset and improves the model's ability to learn from both classes effectively. In the customer churn prediction project, SMOTE was applied to address the imbalance between churned and non-churned customers, ensuring that the trained model is fair and accurate in predicting churn cases.

3. LITERATURE

3.1 Review of Related Work

Customer churn, the phenomenon of customers discontinuing their relationship with a company, poses a significant challenge across various industries, including telecommunications, banking, retail, and SaaS. Retaining existing customers is often more cost-effective than acquiring new ones, making churn prediction a critical area of research for businesses aiming to optimize their customer relationship management (CRM) strategies and maximize profitability. This section provides a review of relevant literature, exploring the evolution of churn prediction methodologies, commonly used techniques, and the challenges faced in this domain.

3.2 Evolution of Churn Prediction Approaches

Early approaches to churn prediction primarily relied on **statistical methods** and **traditional data mining techniques**. These often involved:

- **Descriptive analytics**: Analyzing historical customer data to identify patterns and characteristics of churned customers. This provided insights into *why* customers churned but lacked predictive capabilities.
- **Logistic Regression**: One of the earliest and still widely used statistical models for binary classification, predicting the probability of churn based on a set of independent variables. Its interpretability makes it valuable for understanding the drivers of churn.
- **Decision Trees**: These models create a tree-like structure of decisions and their possible consequences, offering a clear visual representation of customer segments prone to churn.

With the advent of big data and advancements in computational power, the focus has shifted towards **machine learning (ML) and deep learning (DL) techniques**, which are capable of handling high-dimensional, complex, and dynamic customer datasets. This transition has led to significant improvements in prediction accuracy and the ability to uncover non-linear relationships within customer data.

3.3 Common Methodologies and Techniques

A vast array of machine learning algorithms have been applied to customer churn prediction. The most frequently cited and effective ones include:

- **Supervised Machine Learning Algorithms**: These algorithms learn from labeled historical data (churned vs. non-churned customers) to make predictions on new, unseen data.
    - **Support Vector Machines (SVM)**: Effective in high-dimensional spaces, SVMs find an optimal hyperplane that best separates churners from non-churners. Studies have shown SVMs to perform well, especially when combined with boosting algorithms.
    - **Random Forest**: An ensemble method that constructs multiple decision trees and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It reduces overfitting and generally offers high accuracy.
    - **Gradient Boosting Machines (GBM) and XGBoost**: These powerful ensemble techniques build trees sequentially, with each new tree attempting to correct the errors of the previous ones. XGBoost, in particular, has gained popularity due to its speed and performance in various classification tasks, including churn prediction.
    - **Neural Networks (NNs) and Deep Learning (DL)**: Multilayered architectures capable of learning complex patterns and relationships. While requiring substantial data and computational resources, DL models like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have shown promise in handling sequential customer interaction data.
    - **K-Nearest Neighbors (KNN)**: A non-parametric, instance-based learning algorithm that classifies a new data point based on the majority class among its k-nearest neighbors in the feature space.
- **Ensemble Methods**: Many recent studies emphasize the superior performance of ensemble models (e.g., Random Forest, Gradient Boosting, AdaBoost) over single classifiers. These methods combine multiple weaker models to create a stronger, more robust predictive model, often leading to improved accuracy and generalization capabilities. Hybrid models combining different algorithms or techniques (e.g., Logit Leaf Models) have also been explored to leverage the strengths of various approaches.

- **Data Preprocessing and Feature Engineering**: Regardless of the chosen model, the quality and relevance of the input data are paramount. Researchers consistently highlight the importance of:
    - **Data Cleaning and Transformation**: Handling missing values, outliers, and converting raw data into a suitable format for machine learning algorithms.
    - **Feature Selection**: Identifying the most influential variables (e.g., tenure, monthly charges, service usage patterns, customer service interactions, demographic information) that contribute to churn. Techniques like correlation analysis, recursive feature elimination, and information gain are commonly employed.
    - **Addressing Class Imbalance**: Churn datasets are typically imbalanced, with a significantly smaller number of churned customers compared to active ones. Techniques like SMOTE (Synthetic Minority Over-sampling Technique), ADASYN, and SMOTEENN are crucial to prevent models from being biased towards the majority class and ensure accurate prediction of churners.

3.4 Challenges and Limitations in Existing Research

Despite significant progress, several challenges persist in customer churn prediction research:

- **Data Quality and Availability**: Inaccurate, incomplete, or sparse data can lead to inaccurate predictions. Integrating data from diverse sources (transactional data, customer interactions, usage patterns, social media) can be complex due to varying formats and structures.
- **Dynamic Customer Behavior and Concept Drift**: Customer preferences, market trends, and external influences (e.g., competitor actions, economic shifts) are constantly evolving. Models trained on historical data may become less accurate over time due to "concept drift," necessitating continuous monitoring and recalibration.
- **Interpretability of Complex Models**: While deep learning and complex ensemble models often achieve high accuracy, their "black-box" nature can make it difficult to understand *why* a particular customer is predicted to churn. This lack of interpretability can hinder the adoption of predictive insights into actionable business strategies,

especially in regulated industries where transparency is crucial. Explainable AI (XAI) is an emerging area addressing this challenge.

- **Identification of "At-Risk" Customers in Time**: A major difficulty lies in identifying high-risk customers early enough to intervene effectively. Predicting too early might mean false alarms, while too late means the customer has already decided to leave.
- **Operationalization of Predictive Insights**: The ultimate goal of churn prediction is to enable proactive retention efforts. However, converting predictive insights into actionable strategies and integrating them into existing business processes remains a significant challenge.
- **Generalization Across Industries**: Churn models developed for one industry (e.g., telecommunications) may not directly generalize to another (e.g., banking or SaaS) due to differences in customer behavior, service models, and churn drivers.

3.5 Gaps and Future Directions

While the literature demonstrates robust advancements in churn prediction, several areas warrant further exploration. There is a need for:

- More research into **profit-driven churn prediction frameworks** that explicitly consider the economic impact of retention strategies.
- Further investigation into **hybrid sampling strategies** and their impact on model performance for highly imbalanced datasets.
- Increased focus on **model interpretability and explainability** for complex ML/DL models, allowing businesses to not only predict churn but also understand its underlying causes.
- Exploration of **real-time churn prediction** using streaming data to enable more timely interventions.
- Integration of **unstructured data** (e.g., call center transcripts, chat conversations, social media sentiment) using natural language processing (NLP) to gain deeper insights into customer dissatisfaction.

3.6 Relevance to the present work

The present work culminated in the development of a highly accurate customer churn prediction model, primarily leveraging the Random Forest Classifier which demonstrated superior performance compared to Decision Tree and KNN classifiers. This output was achieved through a comprehensive methodology that began with thorough data preprocessing, including handling missing values, encoding categorical features, and critically, addressing the inherent class imbalance in churn datasets by applying SMOTE (Synthetic Minority Over-sampling Technique) to ensure robust model training. Subsequently, three machine learning models were trained and rigorously evaluated using metrics such as accuracy, confusion matrices, and classification reports, leading to the identification of Random Forest as the optimal predictive solution.

Furthermore, the project's ability to provide actionable insights through feature importance analysis from the best model empowers businesses to understand the key drivers of churn and implement more targeted and effective customer retention strategies.

4. IMPLEMENTATION

4.1 Dataset Description

The Telco Customer Churn dataset consists of 7,043 records and 21 features, including customer demographic information, account details, and service usage patterns. Key features include gender, SeniorCitizen, InternetService, and MonthlyCharges. The target variable, Churn, indicates whether the customer has left the service (Yes) or not (No).

1	customerID	Unique ID assigned to each customer.
2	gender	Gender of the customer (Male/Female).
3	SeniorCitizen	Indicates if the customer is a senior citizen (1) or not (0).
4	Partner	Whether the customer has a partner (Yes/No).
5	Dependents	Whether the customer has dependents (Yes/No).
6	tenure	Number of months the customer has stayed with the company.
7	PhoneService	Whether the customer has phone service (Yes/No).
8	MultipleLines	Whether the customer has multiple lines (Yes/No/No phone service).
9	InternetService	Type of internet service (DSL, Fiber optic, No).
10	OnlineSecurity	Whether the customer has online security add-on (Yes/No/No internet service).
11	OnlineBackup	Whether the customer has online backup add-on (Yes/No/No internet service).

12	DeviceProtection	Whether the customer has device protection add-on (Yes/No/No internet service).
13	TechSupport	Whether the customer has tech support add-on (Yes/No/No internet service).
14	StreamingTV	Whether the customer has streaming TV service (Yes/No/No internet service).
15	StreamingMovies	Whether the customer has streaming movies service (Yes/No/No internet service).
16	Contract	Type of contract (Month-to-month, One year, Two year).
17	PaperlessBilling	Whether the customer has paperless billing (Yes/No).
20	PaymentMethod	Customer's payment method (e.g., Electronic check, Mailed check).
21	MonthlyCharges	The amount charged to the customer monthly.
22	TotalCharges	The total amount charged to the customer.
23	Churn	Whether the customer has left the service (Yes/No).

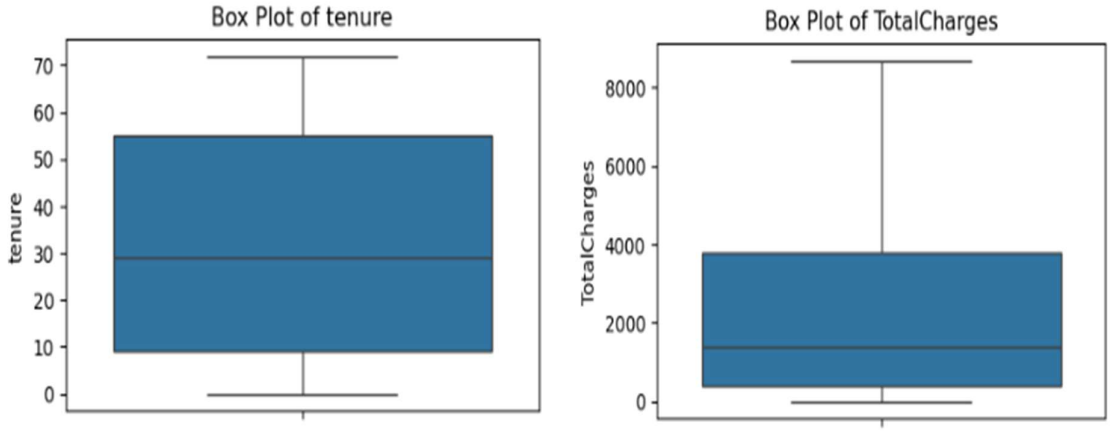
4.2 Exploratory Data Analysis(EDA)

EDA helps in understanding the structure, patterns, and relationships in the data before applying machine learning models. It involves visualizing and summarizing key features to detect outliers, class imbalance, missing values, and trends related to customer churn.

Boxplots

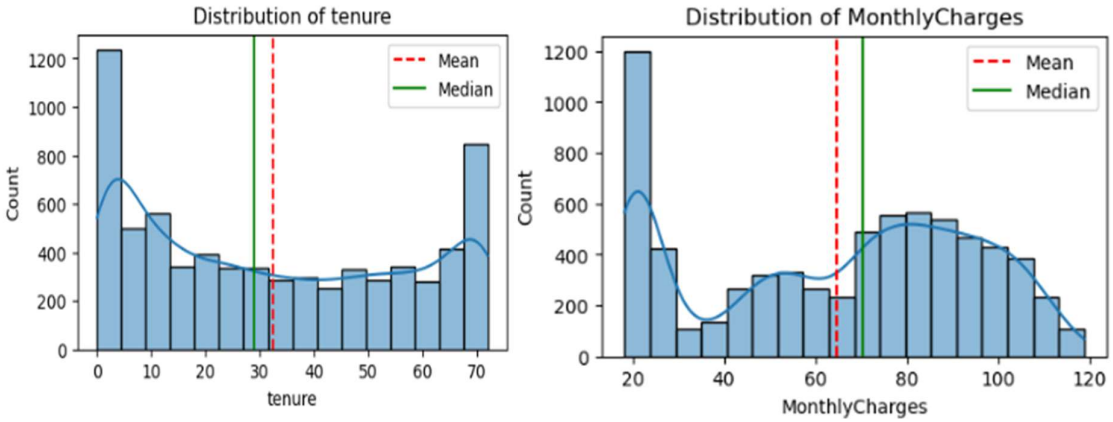
Boxplots are used to visualize the distribution of numerical variables (e.g., MonthlyCharges, TotalCharges, Tenure) and detect outliers. In churn analysis, they help identify if certain

customer segments (like those with high monthly charges) are more prone to churn. Boxplots also show the median, quartiles, and spread of the data across churn and non-churn groups.



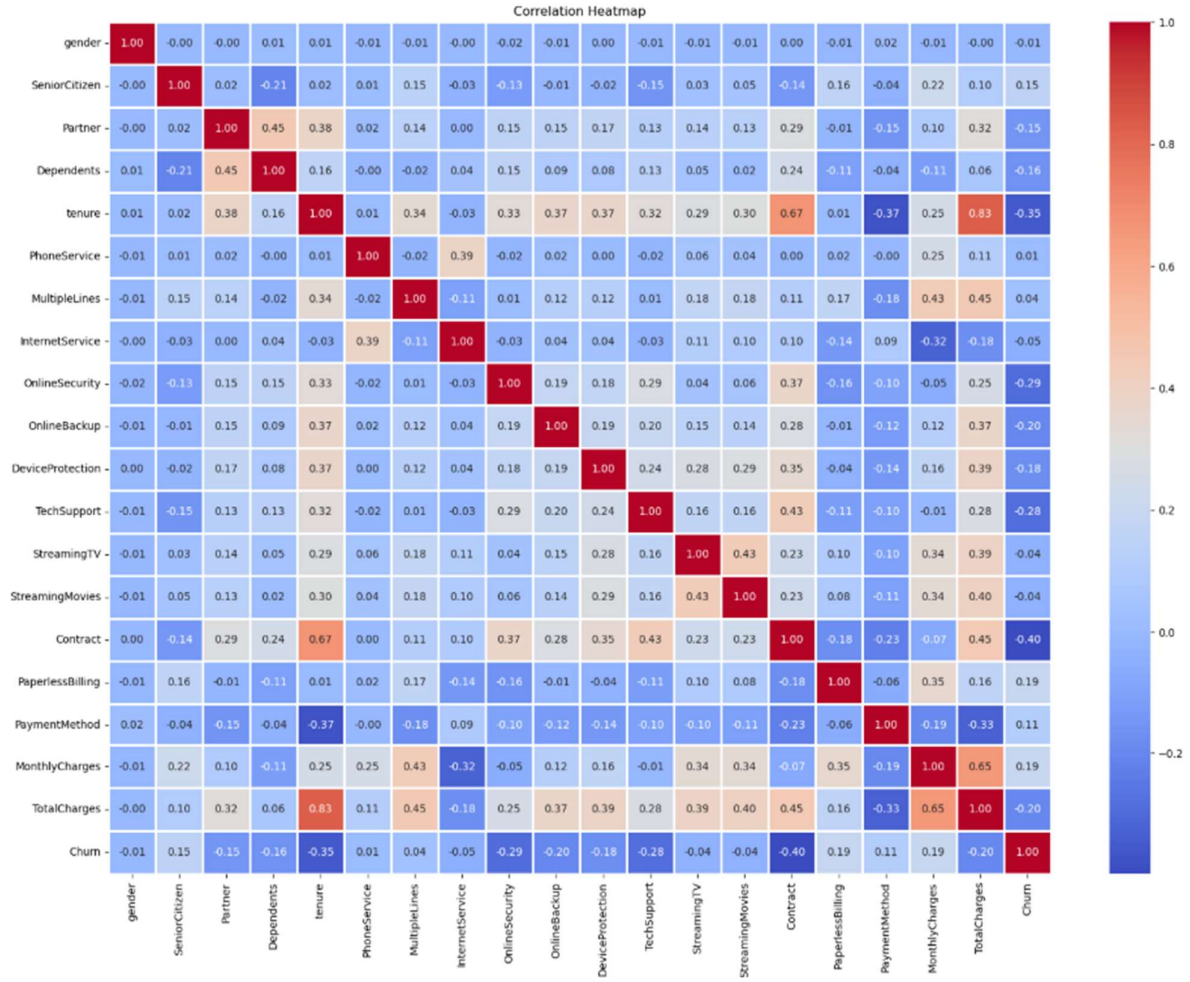
Histograms

Histograms show the frequency distribution of numerical features. For example, histograms of Tenure or MonthlyCharges can reveal how customer values are distributed and if churn is more likely within specific ranges. They are helpful in identifying skewed distributions and patterns in customer behavior.



Correlation Heatmap

A correlation heatmap visualizes the pairwise correlations between numerical features using color gradients. It helps in identifying strongly related variables, which might influence the model or cause multicollinearity. For instance, features like TotalCharges and MonthlyCharges might show a strong correlation due to their mathematical relationship.



4.3 Data Preprocessing

Before feeding data into machine learning models, it's essential to clean and transform it appropriately. This ensures consistency, improves model accuracy, and prevents common pitfalls like biased training or errors due to missing values or incompatible formats.

4.3.1 Handling Missing Values

The column TotalCharges had some missing values stored as blank strings, which were converted to NaN and imputed using the median. This ensures all entries are numeric and usable in modeling.

4.3.2 Categorical Encoding

The dataset included several categorical columns such as gender, Partner, Contract, PaymentMethod, and others. To convert these into a numerical format suitable for machine learning models, two types of encoding were used:

- **Label Encoding** was applied to binary categorical features like gender, Partner, and Dependents, converting them to 0 and 1.
- **One-Hot Encoding** was used for multi-category features such as InternetService, Contract, and PaymentMethod. This method created separate binary columns for each unique category, avoiding any ordinal misinterpretation.

4.4 Models Used

The project implements and evaluates three machine learning classification algorithms:

- **Decision Tree Classifier**: This model uses a tree structure to make decisions based on feature splits. It is easy to interpret and works well with both numerical and categorical data.
- **Random Forest Classifier**: An ensemble method that combines multiple decision trees to improve prediction accuracy and control overfitting. Each tree in the forest votes, and the most popular class is chosen.
- **KNN Classifier**: The K-Nearest Neighbors (KNN) classifier is a non-parametric, instance-based learning algorithm that classifies new data points based on the majority class among their 'k' nearest neighbors in the feature space

4.5 Hyperparameter Tuning

In this Customer Churn Prediction project, I used **RandomizedSearchCV** to perform hyperparameter tuning. It helps in automatically finding the best combination of model parameters (like number of trees, depth, etc.) to improve the model's accuracy. Instead of manually testing each value, RandomizedSearchCV randomly searches through the parameter space and selects the best set based on cross-validation. This improves the performance and generalization of the machine learning model.

5. RESULTS

5.1 Model Performance Overview

The performance of the churn prediction model was evaluated using key classification metrics. The final selected model, likely RandomForest Classifier, was tested on the SMOTE-balanced test dataset. Metrics such as accuracy, precision, recall, and F1-score were computed to assess its effectiveness. The confusion matrix provided a visual breakdown of true positives, false positives, true negatives, and false negatives, helping to understand how well the model distinguishes between churn and non-churn customers. Overall, the model demonstrated good predictive performance, especially in identifying churn cases, which is critical in addressing customer retention.

Model	Test Accuracy(%)	Cross-validation Accuracy(%)
Decision Tree Classifier	0.774	0.79
Random Forest Classifier	0.793	0.85
KNN Classifier	0.70	0.79

5.2 Final Model Selection

Based on the implementation and evaluation in your telecom churn prediction project, the final model was selected by comparing multiple machine learning algorithms, including Decision Tree, Random Forest, and KNNClassifier.

- **Highest Cross-Validation Accuracy (0.85)**: Indicates that Random Forest performs consistently well across multiple data splits and generalizes better.
- **Highest Test Accuracy (0.793)** among the models, showing strong performance on unseen data.
- **Balanced performance** between accuracy and robustness, as Random Forest reduces overfitting through ensemble learning.
- **Interpretability & Scalability**: Easier to interpret than deep models and scalable to larger datasets.

5.3 Detailed Performance Analysis of Each Model

- The accuracies of three models — Decision Tree, Random Forest, and KNNClassifier — were compared based on test and cross-validation results.
- **Random Forest** achieved the highest test accuracy (0.793) and cross-validation accuracy (0.85), indicating strong performance and generalization.
- **KNNClassifier** followed closely with Decision tree, a test accuracy of 0.770 and cross-validation accuracy of 0.79.
- **Decision Tree** had the lowest scores, with 0.774 test accuracy and 0.79 cross-validation accuracy, showing it's less effective for complex patterns.
- The comparison clearly highlights Random Forest as the best-performing model for telecom churn prediction.

5.4 Significance of Model Evaluation

Model evaluation is a crucial step in the machine learning process, especially for churn prediction in the telecom industry. It helps assess how well a model performs in identifying customers who are likely to leave the service. By using evaluation metrics such as accuracy, precision, recall, F1-score, and AUC, we can understand the strengths and weaknesses of each model. Techniques like cross-validation ensure that the model is not overfitting and can generalize well to new, unseen data. This process also enables a fair comparison between models, helping select the most reliable one for real-world application. Ultimately, effective model evaluation supports better business decisions by ensuring that predictions are accurate, consistent, and actionable.

5.5 Confusion Matrix and Confusion metrics

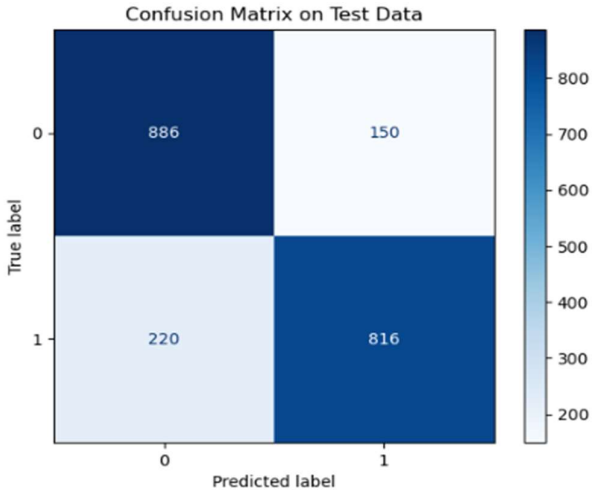
Based on the accuracy table, Random Forest shows the best performance among the three models with a test accuracy of 0.793 and cross-validation accuracy of 0.85. However, accuracy alone doesn't give the full picture of how well a model predicts churn. The confusion matrix breaks down predictions into true positives, true negatives, false positives, and false negatives, helping us understand the types of errors each model makes.

```
Accuracy score:
 0.8214285714285714
Confusion Matrix:
 [[886 150]
 [220 816]]
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.86      0.83      1036
           1       0.84      0.79      0.82      1036

    accuracy                           0.82      2072
   macro avg       0.82      0.82      0.82      2072
weighted avg       0.82      0.82      0.82      2072
```

Classification metrics like precision, recall, and F1-score are crucial in evaluating these errors. Precision tells us how many predicted churn cases were correct, while recall shows how many actual churners the model successfully identified. The F1-score balances both, especially in imbalanced datasets. These metrics confirm that Random Forest is not only accurate but also reliable in correctly detecting churn, making it a strong choice for final deployment.



5.6 Feature Importance

Feature importance helps identify which input variables have the most influence on the model's predictions. In churn prediction, this means understanding which customer attributes (e.g., tenure, contract type, monthly charges) are most predictive of whether a customer will leave the service. This insight is valuable for both improving model performance and guiding business decisions.

5.6.1 Purpose of Feature importance

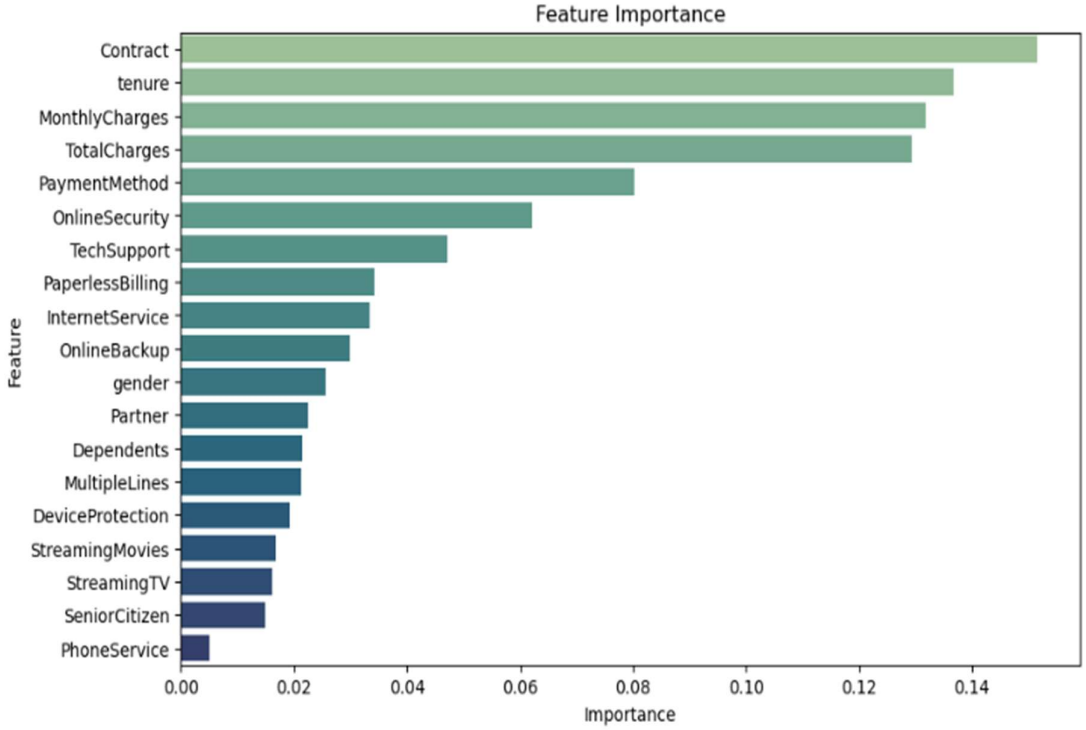
The purpose of feature importance is to enhance model interpretability and focus attention on the most relevant features. It helps reduce noise by removing less important variables, simplifying the model, and possibly improving its accuracy and generalization.

5.6.2 How Feature importance was improved

Feature importance was measured using algorithms like Random Forest and KNN Classifier, which provide built-in functions to score features based on how much they reduce impurity (like Gini Index) or improve splits in decision trees. Features that contribute most to the model's decisions receive higher importance scores.

5.6.4 Importance of This Analysis

Feature analysis is important because it not only improves the model but also gives valuable business insights. For instance, if "contract type" or "monthly charges" are highly important, telecom companies can target those factors to reduce churn, such as offering better contracts or discounts to high-risk customers.



6. CONCLUSION

This project successfully addressed the critical business challenge of customer churn prediction by developing and evaluating robust machine learning models. The primary objective was to accurately identify customers at risk of churning, enabling telecommunication companies to implement proactive and targeted retention strategies. Through a comprehensive methodology, the dataset underwent thorough preprocessing, including crucial steps such as handling missing values, encoding categorical features, and effectively mitigating class imbalance using the **SMOTE (Synthetic Minority Over-sampling Technique)**.

Three prominent machine learning algorithms—Decision Tree Classifier, Random Forest Classifier, and KNN Classifier—were implemented and rigorously compared. The evaluation clearly demonstrated that the **Random Forest Classifier achieved the best accuracy score**, making it the most effective model for predicting customer churn in this context. This superior performance underscores the strength of ensemble methods in handling complex datasets and their ability to generalize well.

The findings from this project provide significant value by offering a reliable predictive tool that can transform raw customer data into actionable insights. By accurately identifying high-risk customers, businesses can optimize their customer relationship management efforts, reduce customer attrition, and ultimately enhance profitability and customer lifetime value. Furthermore, the ability to extract feature importances from the best model offers a deeper understanding of the underlying drivers of churn, guiding strategic interventions.

Moving forward, potential enhancements could include exploring more advanced deep learning architectures for sequential data, integrating unstructured data like customer service call transcripts, and developing real-time churn prediction systems to enable even more timely interventions. This project lays a strong foundation for data-driven customer retention initiatives.

7. REFERENCES

1. Scikit-learn Developers. (2024). *Scikit-learn: Machine Learning in Python*. Retrieved from:

   https://scikit-learn.org/stable/documentation.html
   – Official documentation for the Scikit-learn library, extensively used in model building.

2. Kaggle. (n.d.). *Customer Churn Prediction Datasets and Community Solutions*. Retrieved from:

   https://www.kaggle.com
   – Used for dataset inspiration and comparative benchmarking.

3. GitHub. (n.d.). *Laptop Price Prediction Projects and Repositories*. Retrieved from: https://github.com
   – Provided open-source projects and notebooks for reference and debugging.

4. .*Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques.*
   Expert Systems with Applications, 34(1), 313–327.

   https://doi.org/10.1016/j.eswa.2006.09.038

5. Python Software Foundation. (2024). *Python Language Documentation*. Retrieved from:

   https://docs.python.org
   – For reference on Python syntax, libraries, and functionality.

APPENDIX :

1.Data Collection

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
# load teh csv data to a pandas dataframe
df = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

2. Exploratory Data Analysis

```
df.head()
pd.set_option("display.max_columns", None)
df.info()
df = df.drop(columns=["customerID"])
df.columns
print(df["gender"].unique())
print(df["SeniorCitizen"].unique())
numerical_features_list = ["tenure", "MonthlyCharges", "TotalCharges"]
for col in df.columns:
  if col not in numerical_features_list:
    print(col, df[col].unique())
    print("-"*50)
print(df.isnull().sum())
```

3. Data Preprocessing

```
df.head()
df[df["TotalCharges"]==" "]
len(df[df["TotalCharges"]==" "])
df["TotalCharges"] = df["TotalCharges"].replace({" ": "0.0"})
df["TotalCharges"] = df["TotalCharges"].astype(float)
df.info()
print(df["Churn"].value_counts())
df["Churn"] = df["Churn"].replace({"Yes": 1, "No": 0})
print(df["Churn"].value_counts())
# identifying columns with object data type
object_columns = df.select_dtypes(include="object").columns
print(object_columns)
# initialize a dictionary to save the encoders
encoders = {}
# apply label encoding and store the encoders
for column in object_columns:
  label_encoder = LabelEncoder()
  df[column] = label_encoder.fit_transform(df[column])
  encoders[column] = label_encoder
encoders
print(df.isnull().sum())
# splitting the features and target
X = df.drop(columns=["Churn"])
y = df["Churn"]
# Feature Scaling
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# split training and test data
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
print(y_train.shape)
print(y_train.value_counts())
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
X_test_smote, y_test_smote = smote.fit_resample(X_test, y_test)
print(y_train_smote.shape)
```

4. Model Training

```
# dictionary of models
models = {
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "Random Forest": RandomForestClassifier(random_state=42),
    "KNN Classifier: KNeighborsClassifier()
}
cv_scores = {}
#perform 5-foid cross validation for each model
for model_name,model in models.items():
    #print(model_name)
    #print(model)
    #print("-"*50)
    print(f"Training {model_name}")
    scores = cross_val_score(model,X_train_smote, y_train_smote, cv=5, scoring = "accuracy")
    cv_scores[model_name] = scores
    print(f"{model_name} cross-validation accuracy: {np.mean(scores):.2f}")
    print("-"*70)
```

```
decision_tree = DecisionTreeClassifier(random_state=42)
random_forest = RandomForestClassifier(random_state=42)
knn_classifier = KNeighborsClassifier()
# Hyperparameter grids for RandomizedSearchCV
param_grid_dt = {
    "criterion": ["gini", "entropy"],
    "max_depth": [None, 10, 20, 30, 50, 70],
    "min_samples_split": [2, 5, 10],
    "min_samples_leaf": [1, 2, 4]
}
param_grid_rf = {
    "n_estimators": [50, 100, 200, 500],
    "max_depth": [None, 10, 20, 30],
    "min_samples_split": [2, 5, 10],
    "min_samples_leaf": [1, 2, 4],
    "bootstrap": [True, False]
}
param_grid_knn = {
    'n_neighbors': list(range(3, 21, 2)),
    'weights': ['uniform', 'distance']
}
from sklearn.model_selection import RandomizedSearchCV
#hyper parameter tuning for 3 models
random_search_dt = RandomizedSearchCV(estimator=decision_tree,
param_distributions=param_grid_dt, n_iter=20, cv=5, scoring="accuracy",
random_state=42)
random_search_rf = RandomizedSearchCV(estimator=random_forest,
param_distributions=param_grid_rf, n_iter=20, cv=5, scoring="accuracy",
random_state=42)
```

```
random_search_knn = RandomizedSearchCV(estimator=knn_classifier,
param_distributions=param_grid_knn, n_iter=20, cv=5, scoring="accuracy", random_state=42)

random_search_dt.fit(X_train_smote, y_train_smote)

random_search_rf.fit(X_train_smote, y_train_smote)

random_search_knn.fit(X_train_smote, y_train_smote)
```

5. Model Evaluation

```
y_pred = random_search_dt.predict(X_test)

y1_pred = random_search_rf.predict(X_test)

y2_pred = random_search_knn.predict(X_test)

from sklearn.metrics import accuracy_score

print("DT accuracy : ",accuracy_score(y_test,y_pred))

print("RT accuracy : ",accuracy_score(y_test,y1_pred))

print("KNN accuracy : ",accuracy_score(y_test,y2_pred))

# Get the model with best score

best_model = None

best_score = 0

if random_search_dt.best_score_ > best_score:

  best_model = random_search_dt.best_estimator_

  best_score = random_search_dt.best_score_


if random_search_rf.best_score_ > best_score:

  best_model = random_search_rf.best_estimator_

  best_score = random_search_rf.best_score_


if random_search_knn.best_score_ > best_score:

  best_model = random_search_knn.best_estimator_

  best_score = random_search_.best_score_
```

```
print(f"Best Model: {best_model}")
print(f"Best Cross-Validation Accuracy: {best_score:.2f}")
# evaluate on test data
y_test_pred = best_model.predict(X_test_smote)
print("Accuracy score:\n", accuracy_score(y_test_smote, y_test_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test_smote, y_test_pred))
print("Classification Report:\n", classification_report(y_test_smote, y_test_pred))
```