CUSTOMER CHURN ANALYSIS

PHASE 2

AREA OF PROJECT:
MACHINE LEARNING

TEAM MEMBERS

TEAM NUMBER-44

TANEESHKA NAGANATH REDDY
01JCE21CS116

MADHUSUDHAN
01JCE21CS060

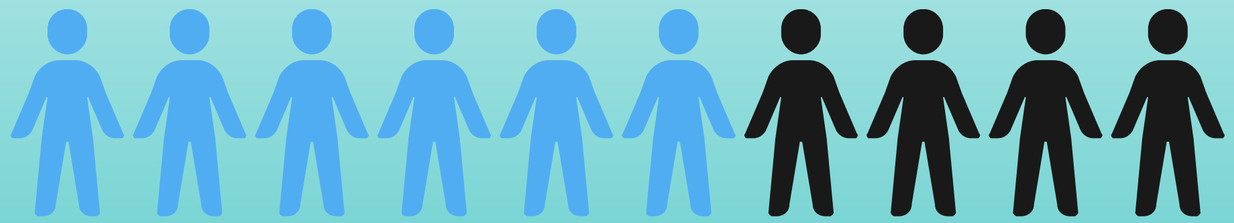
HARSHA N C
01JST22UCS410

SANKET
01JST22UCS430

PROJECT GUIDE

ASST.PROF BINDIYA A R
DEPARTMENT OF CSE
JSS STU
MYSURU-570006

TABLE OF CONTENTS



01-INTRODUCTION

WHAT IS CHURN?

Churn is a measurement of the percentage of accounts that cancel or choose not to renew their subscriptions.

It basically measures the amount of customers, accounts, contracts, bookings, etc. that a business has lost over a period of time.

Managing customer churn is one major challenge companies face, Especially those offering subscription-based services.

Any organization’s primary motive should be satisfying customers and retaining existing customers.

02- LITERATURE REVIEW

THE TELECOM INDUSTRY

- Customers frequently switch operators due to various factors.
- Predictive factors are call patterns, data usage, service quality perceptions, pricing strategies, customer service interactions, and network coverage.

PREVIOUS STUDIES ON CUSTOMER CHURN

- Ning Lu suggests using boosting algorithms to improve customer churn prediction models in which customers are clustered based on the boosting algorithm's weight. A high-risk customer cluster was found. ***Logistic regression (LR)*** is used as a learner, and each cluster has a churn prediction model. The results showed that boosting algorithm separates churn data better than a single logistic regression model.[1]

- M.A.H. Farquad present an avenue to overwhelm the fault of general SVM that creates a black box model. The author constructed an approach that divides into three phases. In the 1st phase, SVM Recursive Feature Elimination (SVM-RFE) is hired to decrease the feature set. During 2nd phase, dataset with minimal features are extracted and then apply the ***SVM techniques*** to do the classification. In the last phase, rules are extracted manually. After extracting the rules, Naive Bayes is combined with ***Decision tree*** and produce the result.[2]

PREVIOUS STUDIES ON CUSTOMER CHURN

- Edwine et al., have done a comparative analysis of customer churn prediction models in the telecom industry. They have used three best-fit algorithms, namely **KNN, RF, and SVM along with an optimization algorithm for hyperparameter tuning.** They have concluded that the basic versions of these algorithms perform lesser than the amalgamation (RF with grid search optimization algorithm) with a low-ratio undersampling strategy.[3]
- Edvaldo and Olawande have proposed that as of not long ago, con ventional AI strategies (for example, MLP and SVM were effectively utilized for pivot forecast, however with impressive exertion in the design of the preparation boundaries [15]. They were foreseeing even information for client relationships with the executives in finance utilizing **Deep Neural Networks (DNN).** [4]

03 - DESIGN

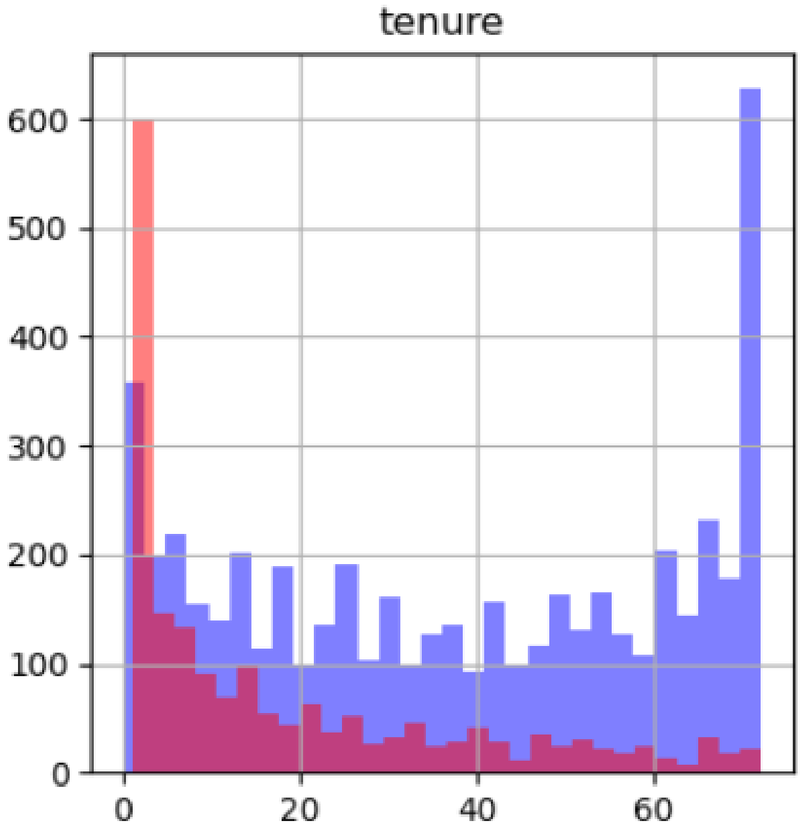
The data set includes information about:

- Customers who left within the last month – the column is called Churn
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents

03 - DESIGN

Data Cleaning:
• Checks for null values in features.
• Drops or replaces null values with mean.

Feature Engineering:
• Plots distributions for numerical and categorical features.
• Uses histograms for numerical features distribution.
• Uses bar charts for categorical features analysis.



03 - DESIGN

One Hot Encoding:

- Represents categorical variables as numerical values in a machine learning model.
- Converts all features into numerical values for machine learning models.

Data Standardization:

- Transforms features by subtracting from mean and dividing by standard deviation.
- Provides equal footing for all features.
- Avoids over-influence of features with larger scales.

03 - DESIGN

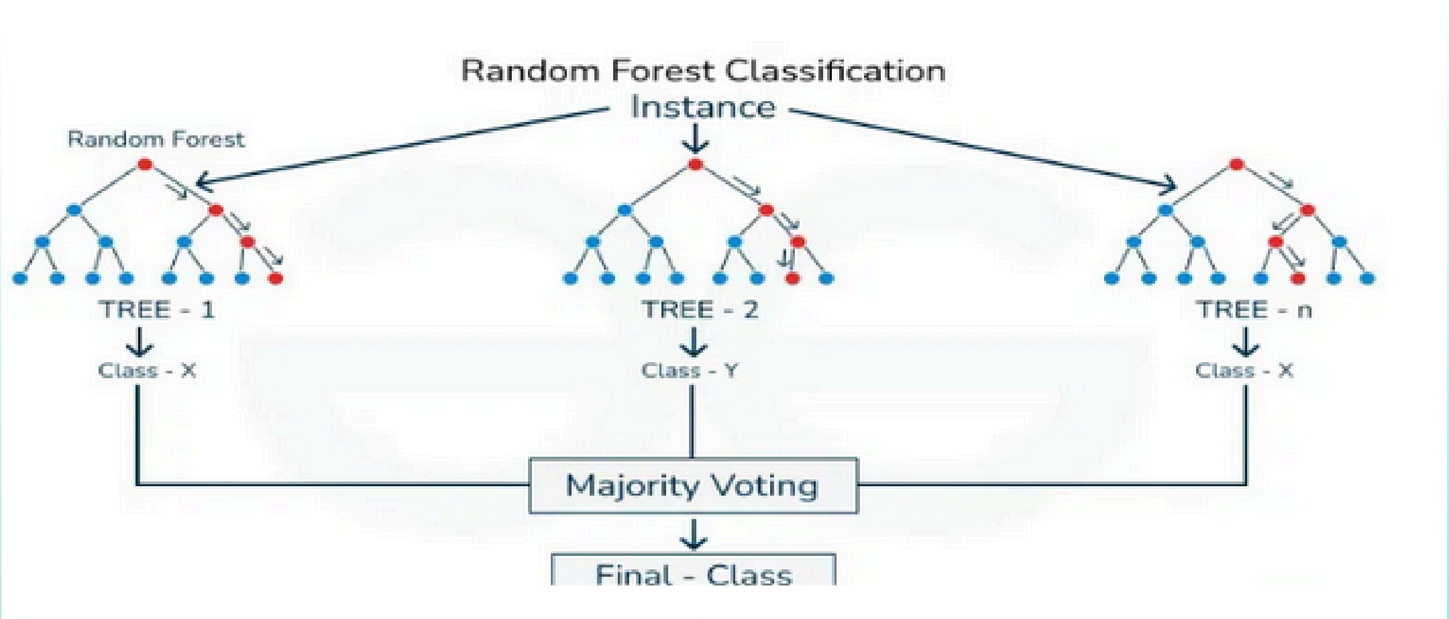
Handling Imbalanced Data in Data Analysis

- SMOTE (Synthetic Minority Oversampling Technique) is used to minimize imbalance classes.
- SMOTE creates new, artificial customers who churn, focusing on the minority class
- SMOTE creates new data points similar but not identical to existing customers.

03 - DESIGN

- **Random Forest Classification :**

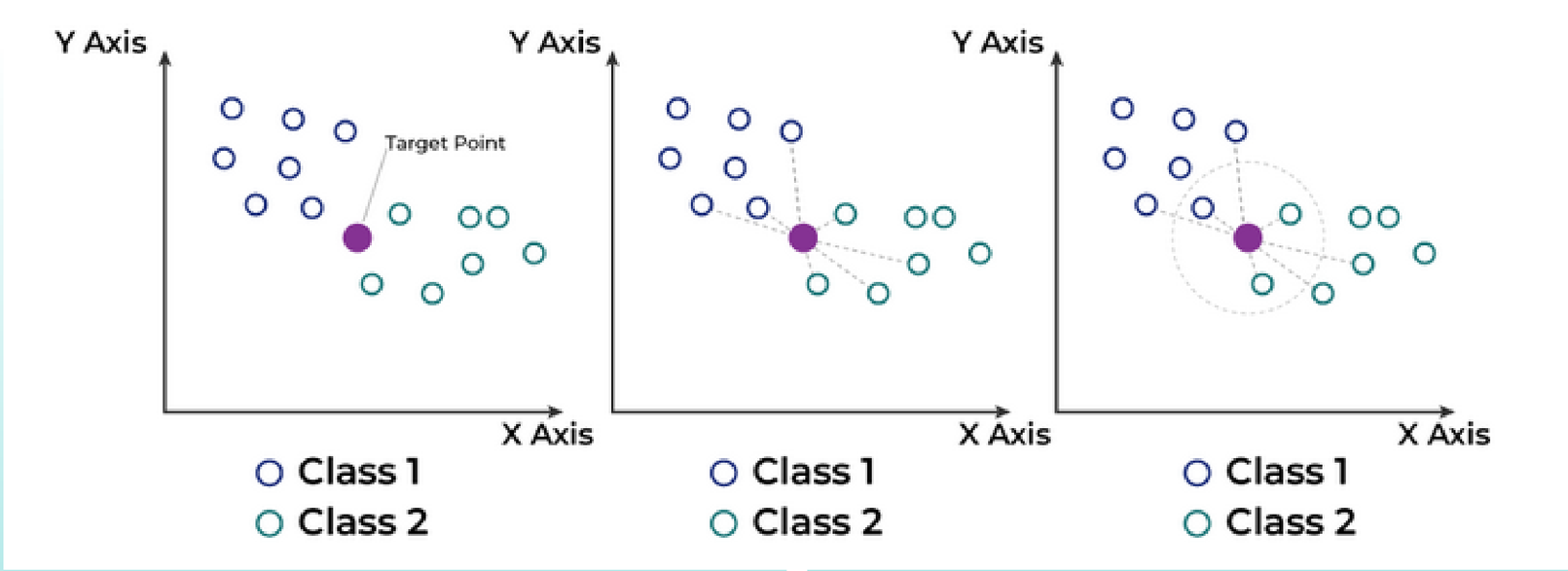
  Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.



03 - DESIGN

- **KNN (K-Nearest Neighbors)**

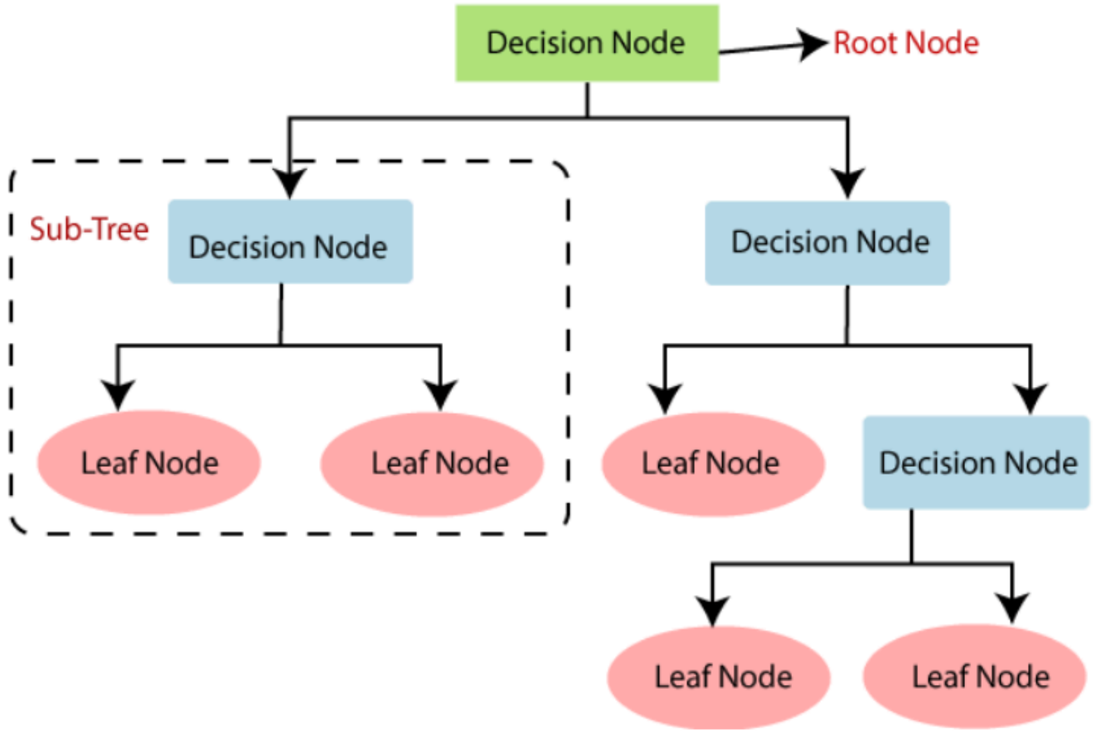
The K-NN algorithm works by finding the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance.



03 - DESIGN

- **Decision Tree:**

The algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.



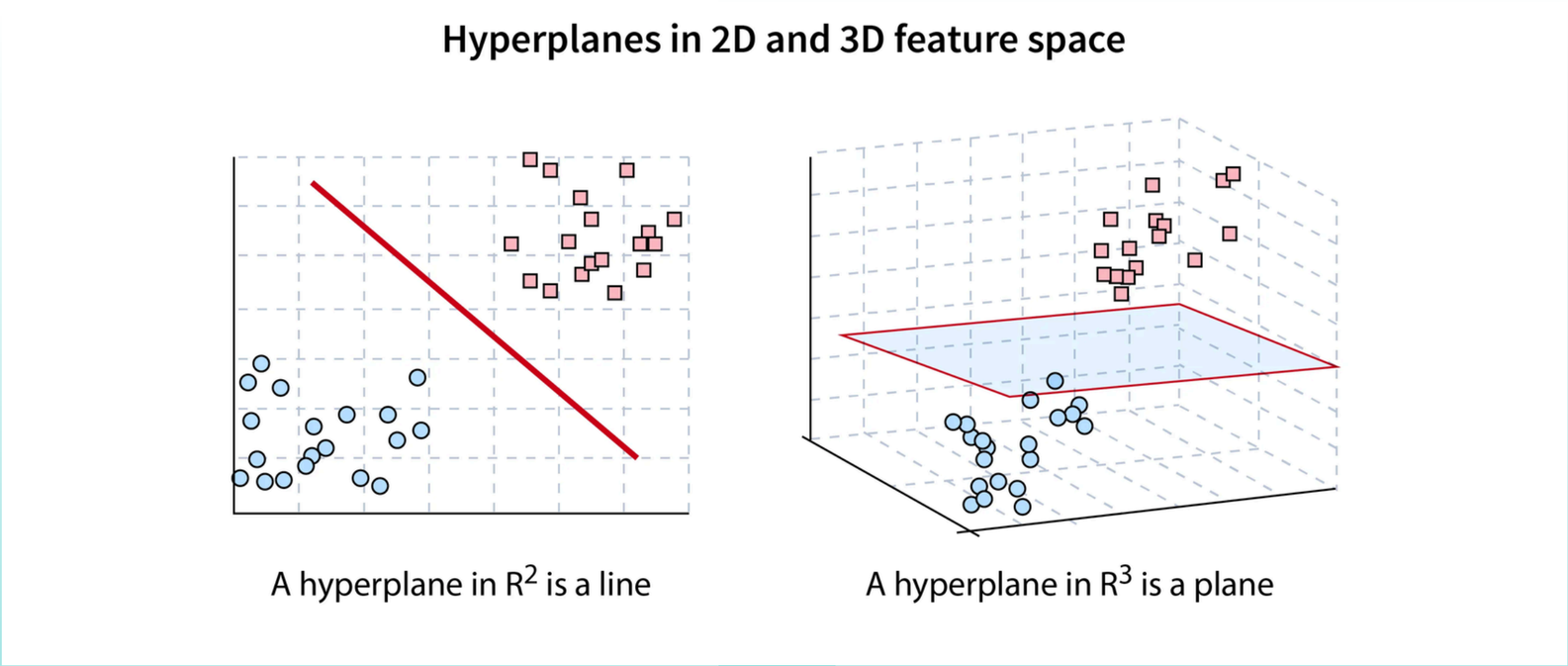
- **Attribute Selection Measures ASM):**

An attribute selection measure is a heuristic for selecting the splitting criterion that. best separates a given data set D of class-labeled training tuples into individual classes.

03 - DESIGN

Support Vector Classifier (SVC):

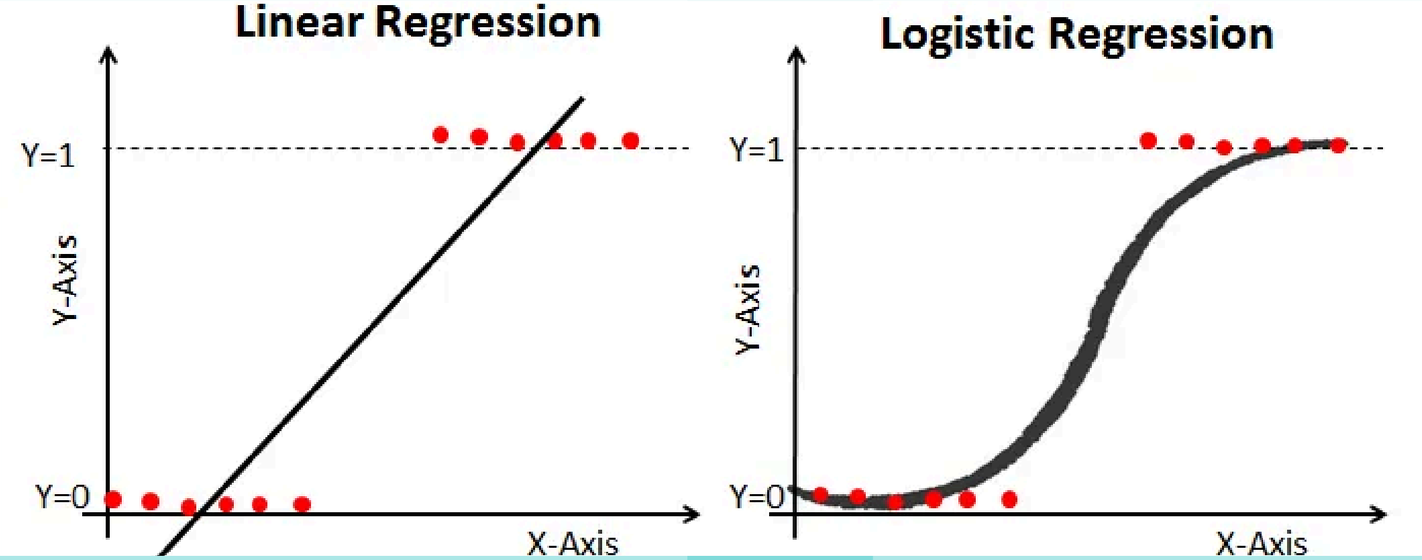
The aim of a support vector machine algorithm is to find the best possible line, or decision boundary, that separates the data points of different data classes. This boundary is called a hyperplane when working in high-dimensional feature spaces.



03 - DESIGN

Logistic Regression

Logistic regression is an algorithm that provides a linear relationship between an independent variable and a dependent variable to predict the outcome of future events. It is a statistical method used in data science and machine learning for predictive analysis.

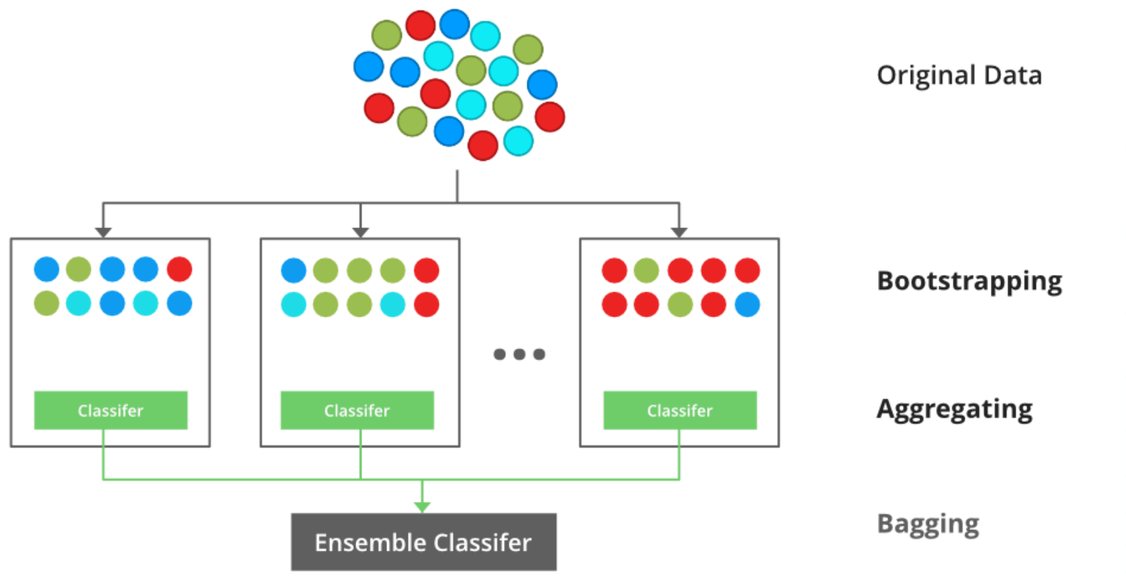


03 - DESIGN

XGBoost:

XGBoost, short for **eXtreme Gradient Boosting,** is a powerful and efficient open-source implementation of the gradient boosting algorithm.

XGBoost is an ensemble learning method that builds a strong predictive model by combining the outputs of several weaker models, typically decision trees. It improves the model by iteratively adding trees, focusing on correcting the errors made by the previous trees.



03 - DESIGN

Model Evaluation Metrics:

•**Classification accuracy:** Calculated by dividing the ratio of correct predictions to the total number of input samples

•**Precision:** Measures the number of correct positive predictions by the number of true positive and false positive predictions.

P= TP/(TP+FP)

•**Recall:** Lower recall and higher precision lead to greater accuracy but may miss a large number of instances.

R =TP/(TP+FN)

03 - DESIGN

Model Evaluation Metrics:

• **F1 Score:** A harmonic mean between recall and precision, indicating the precision and robustness of the classifier.

$$F1 = (2 * P * R) / (P + R)$$

• **ROC:** Stands for Receiver Operating Characteristics, plotting the true positive rate (TPR) vs the false positive rate (FPR) at different classification thresholds

• **AUC:** Area Under the ROC curve It provides a single scalar value that summarizes the performance of the model. The value of AUC ranges from 0 to 1:

04-IMPLEMENTATION

STEP 1] SAVING THE MODEL USING PICKLE

‘pickle’ is a Python module that provides a way to serialize and deserialize Python objects.

Serialization is the process of converting a Python object into a byte stream, which can be saved to a file or sent over a network.

Deserialization is the process of converting the byte stream back into a Python object.

Using pickle, you can save complex data structures such as lists, dictionaries, and even custom objects.

So here we save the model with the highest accuracy (logistic regression) using pickle and extension .sav

STEP 2] FRONTEND USING HTML,CSS AND FLASK

- **Setup:** Flask application (app.py) is set up with necessary libraries and dependencies, including loading the trained machine learning model (lr_model.sav).

- **Home Route (/):** When the user accesses the root URL, the home() function is triggered. This function renders the index.html template, which contains a form for inputting customer data.

- **Prediction Route (/predict):** When the user submits the form, the predict() function is triggered. This function extracts the input values from the form and processes them to match the format expected by the model.

- **Data Processing:** The input values are converted to the appropriate data types (e.g., integer, float, boolean) based on the model's requirements.

- **Model Prediction:** The processed data is then fed into the loaded machine learning model (model) for prediction. The model predicts whether the customer will churn or not.

- **Result Display:** Depending on the prediction result an appropriate message is displayed on the web page.

- **HTML Templates:** The HTML templates (index.html) are used to create the user interface for entering data and displaying the prediction result.

- **Styling:** CSS styles are applied to the HTML elements to improve the visual appeal and user experience.

- **Running the Application:** Finally, the Flask application is run with the python app.py on the terminal , which starts the web server and makes the application accessible through a web browser.

05-TEST CASES AND RESULTS

Model Performance:

Model	Accuracy	Precision (False Class)	Recall (True Class)	F1- Score (False Class)	ROC-AUC
Logistic Regression	0.78	0.88	0.69	0.85	0.75
Support Vector Classifier (SVC)	0.74	0.84	0.56	0.83	0.68
Decision Tree	0.72	0.83	0.57	0.80	0.67
KNN	0.772	0.83	0.51	0.85	0.689
Random Forest	0.79	0.85	0.59	0.86	0.72
XGBoost	0.77	0.88	0.61	0.85	0.72

05-TEST CASES AND RESULTS

Based on the performance metrics provided:
Logistic Regression appears to be the best overall model due to its high accuracy, balanced precision and recall, high F1-scores, and the highest ROC-AUC score. This indicates it has a good balance between sensitivity (recall) and specificity, and it provides a reliable measure of overall performance.

Why Choose Logistic Regression?
High Accuracy: It has one of the highest accuracy rates among the models tested.

Balanced Performance: It has balanced precision and recall, ensuring that both classes are well-represented.

ROC-AUC Score: It has the highest ROC-AUC score, which means it performs well in distinguishing between the churn and non-churn classes.

05-TEST CASES AND RESULTS

Test Case 1:

Input:

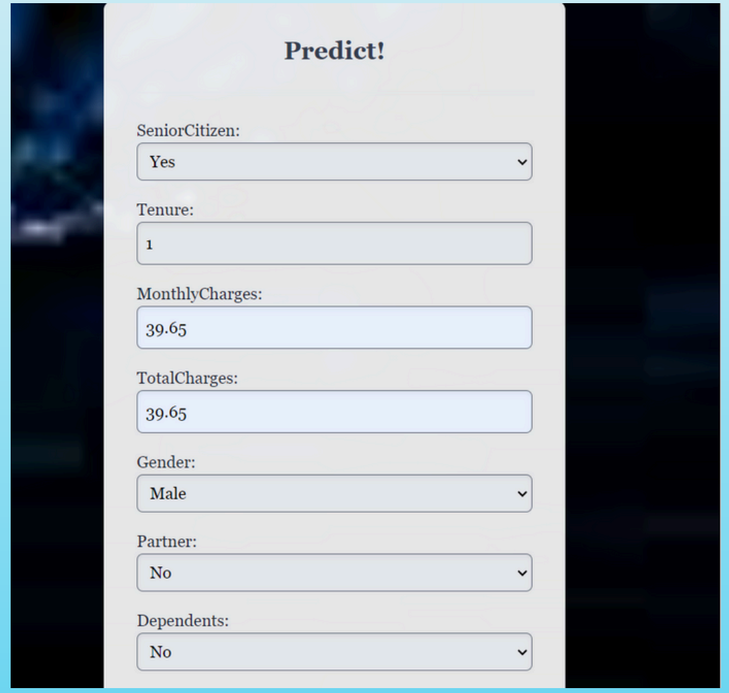
SeniorCitizen: 1
tenure: 1
MonthlyCharges: 39.65
TotalCharges: 39.65
gender: m
Partner: No
Dependents: No
PhoneService: No
MultipleLines: No phone service
InternetService: DSL
OnlineSecurity: Yes
OnlineBackup: yes
DeviceProtection: No
TechSupport: No
StreamingTV: No
StreamingMovies: Yes
Contract: Month-to-month
PaperlessBilling: Yes
PaymentMethod: Electronic

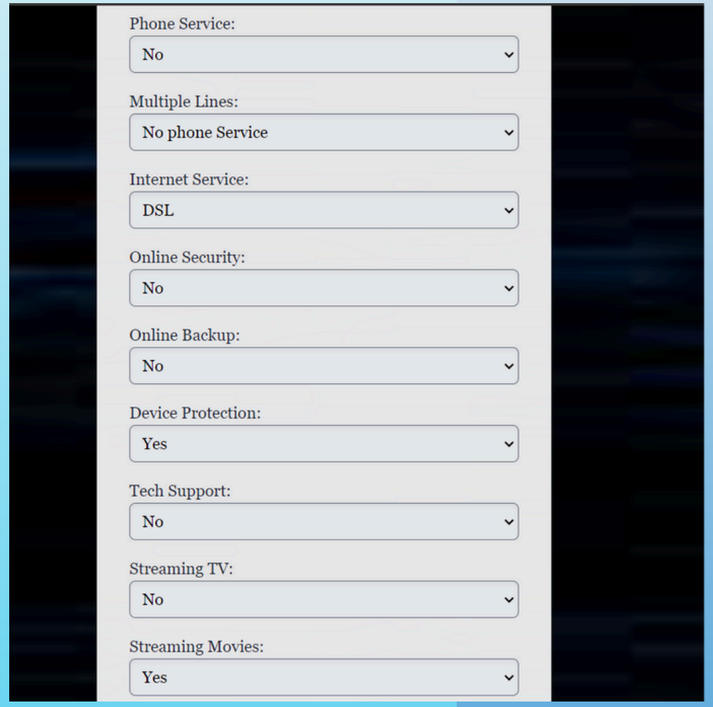
Expected Result:
Customer WILL CHURN

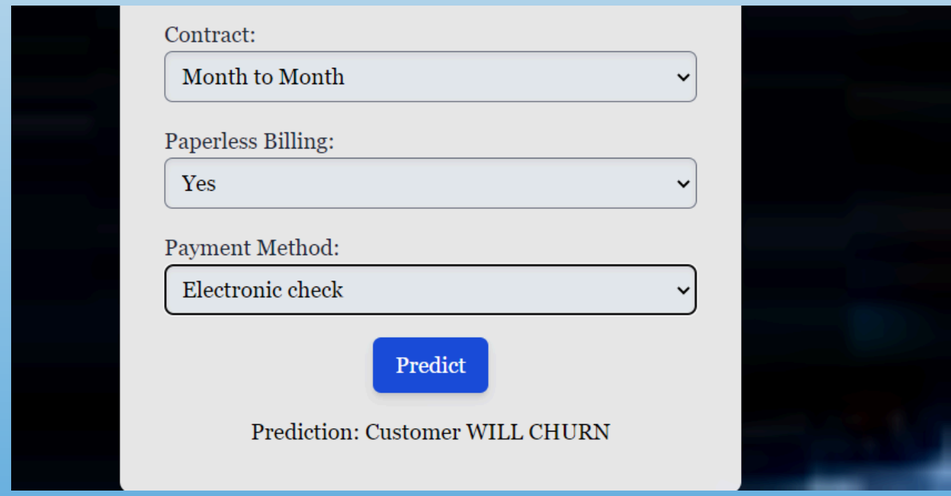
Actual Result:
Customer WILL CHURN

Explanation:
The model correctly predicted that the customer will churn based on the input features.

05-TEST CASES AND RESULTS







Test Case 2:

Input:

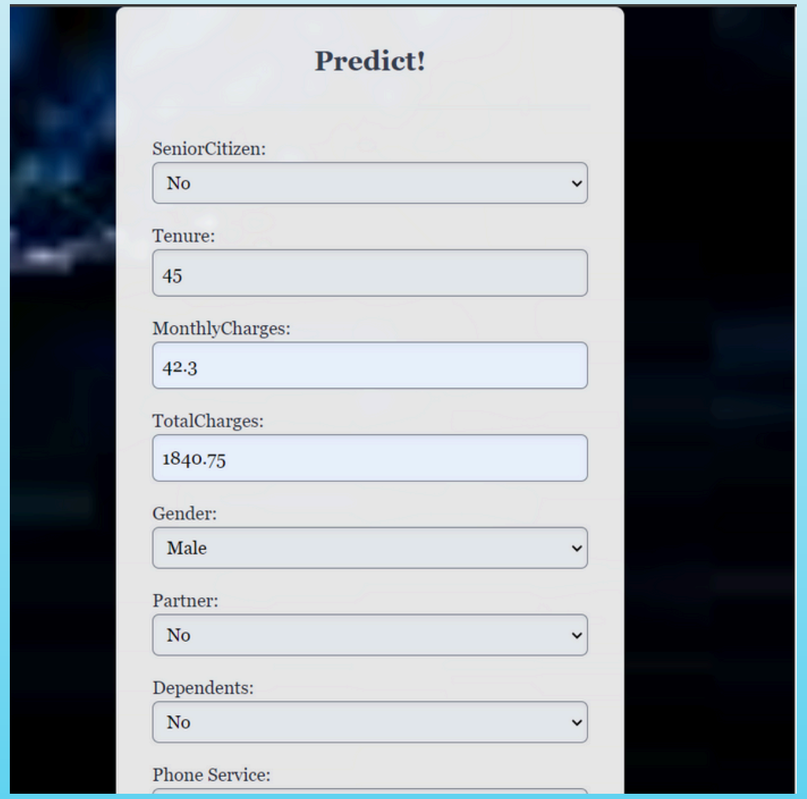
SeniorCitizen: 0
tenure: 45
MonthlyCharges: 42.3
TotalCharges: 1840.75
gender: m
Partner: No
Dependents: No
PhoneService: No
MultipleLines: No phone service
InternetService: DSL
OnlineSecurity: Yes
OnlineBackup: No
DeviceProtection: Yes
TechSupport: Yes
StreamingTV: No
StreamingMovies: No
Contract: One year
PaperlessBilling: No
PaymentMethod: bank transfer

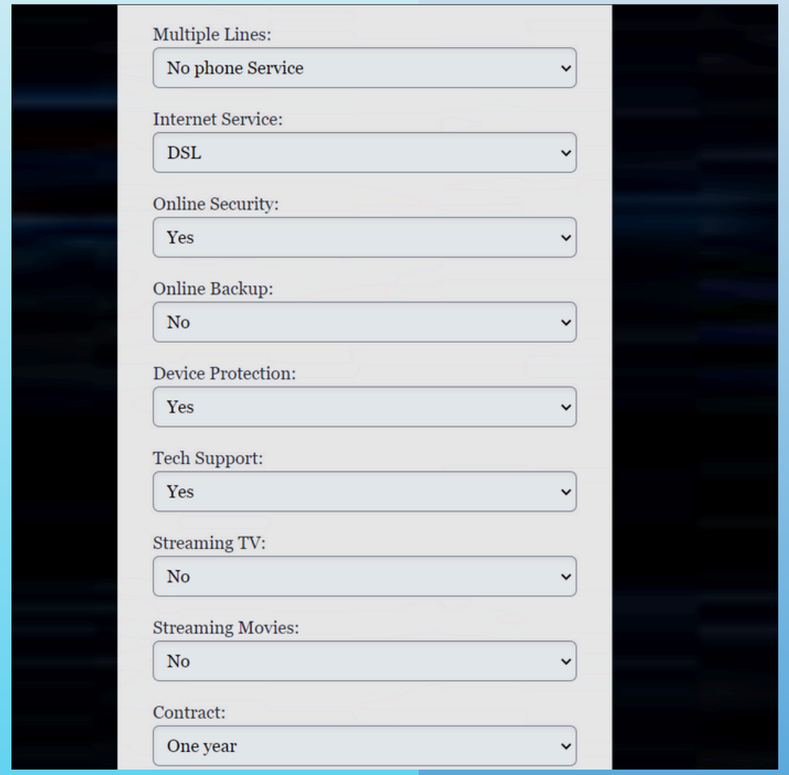
Expected Result:
Customer WILL NOT CHURN

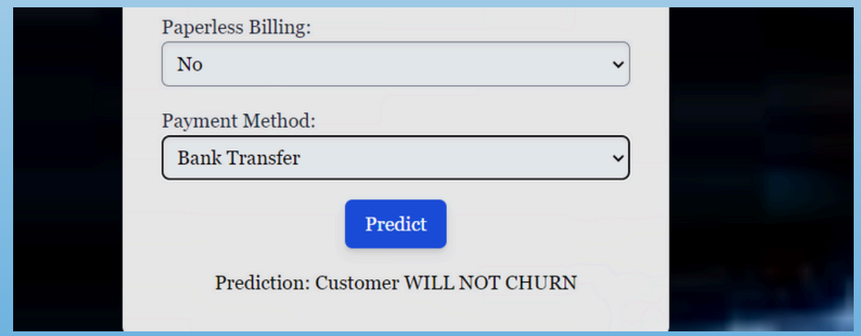
Actual Result:
Customer WILL NOT CHURN

Explanation:
The model correctly predicted that the customer will churn based on the input features.

05-TEST CASES AND RESULTS







06 - FUTURE ENHANCMENTS

- **Feature Engineering:** Exploring additional features or engineered variables could enhance model performance and capture more nuanced patterns in customer behaviour.

- **Model Tuning:** Fine-tuning hyperparameters and optimising model architectures can further improve predictive accuracy and robustness.

- **Deployment and Monitoring:** Continuous monitoring of model performance post-deployment is crucial for ensuring its effectiveness in real-world scenarios. Implementing mechanisms for model updating and retraining based on incoming data can help maintain predictive accuracy over time.

- **Predicting Exact Churn Time**: Extend the current model to predict not only whether a customer will churn but also the exact time or the estimated time frame when the churn is likely to happen. This could involve using time-to-event analysis techniques such as survival analysis.

07-REFERENCES

[1] - Ning Lu, Hua Lin, Jie Lu, Guangquan Zhang (May 2014.)“A Customer Churn Prediction Model in Telecom Industry Using Boosting”, IEEE Transactions on Industrial Informatics, vol. 10, no.2
https://ieeexplore.ieee.org/abstract/document/6329952

[2] - Farquad, H. &Vadlamani, Ravi &Surampudi, Bapi. (2014). Churn Prediction using Comprehensible Support Vector Machine: an Analytical CRM Application. Applied Soft Computing. 19. 10.1016/j.asoc.2014.01.031
https://www.sciencedirect.com/science/article/abs/pii/S1568494614000507

07-REFERENCES

[3] - Nabahirwa Edwine, Wenjuan Wang, Wei Song, Denis Ssebuggwawo, (2022) Detecting the risk of customer churn in the telecom sector: a comparative study, Math. Probl Eng. 2022. Article ID 8534739, 16 pages. https://onlinelibrary.wiley.com/doi/10.1155/2022/8534739

[4] - Edvaldo Domingos, Blessing Ojeme, Olawande Daramola,(2021) Experimental analysis of hyperparameters for deep learning-based churn prediction in the banking sector, Computation 9 (3) 34. https://www.mdpi.com/2079-3197/9/3/34

THANKYOU