AI LAB Mini Project Report

Customer Churn Analysis using SVM , PCA + Clustering, XGBoost, and Artificial Neural Networks

Team Members

1. Tvisha Vedant (221071075)
2. Kamakshi Ramamurthy (221071053)
3. Anushka Yadav (221071081)
4. Sakshi Khanduri (221071033)

Faculty:Tabassum Kazi Ma'm and S.C.Shrawane Ma'm (Class Teacher)
Sem: 6
Branch: CS
Batch: D

Abstract

Customer churn is a major concern for subscription-based businesses. This project explores three distinct approaches for churn analysis: Principal Component Analysis (PCA) followed by Clustering, XGBoost Classification, and Artificial Neural Networks (ANNs). By leveraging these methods, the aim is to segment customers, predict churn likelihood, and identify contributing factors. The analysis enables companies to implement data-driven retention strategies, reduce customer attrition, and enhance loyalty.

Table of Contents

1. Introduction

2. Set of Deliverables

3. Features and Functionalities

4. Details of the Project
    - Data processing
    - Explanation of Algorithms
        - SVM
        - XGBoost
        - Supervised Learning for predicting churned users
        - PCA + Clustering
        - Artificial Neural Networks (ANNs)

    - System Architecture and Workflow

5. Results and Observations

6. Conclusion

1. Introduction

Customer churn impacts the profitability of a business significantly. Retaining existing customers is not only more cost-effective than acquiring new ones, but also enhances brand loyalty and long-term revenue. Through advanced machine learning techniques, businesses can proactively identify churn risks and develop intervention strategies.

Objectives:

- Reduce customer churn through early prediction.

- Identify patterns in customer behavior.

- Segment customers using clustering.

- Highlight influential churn factors using feature importance.

- Develop a robust prediction model using deep learning.

2. Set of Deliverables

- Supervised Learning for prediction and Dimensionality reduction using PCA.
- Clustering of customer segments using K-Means or Hierarchical Clustering after PCA.
- Classification model using XGBoost to predict churn.
- ANN-based model to capture non-linear patterns in customer behavior.
- Performance comparison of the models using metrics like accuracy, precision, recall, and F1-score.
- Visualizations and dashboards for customer segmentation and churn probability.

3. Features and Functionalities

1. **SVM (Support Vector Machine)**

Goal of SVM in Customer Churn Analysis

- **Classify customers** into two categories: **churned** vs **non-churned**.
- **Identify patterns** in customer behavior that indicate a likelihood to churn.
- **Improve customer retention** by detecting churn early and enabling proactive actions.
- **Maximize model accuracy** while maintaining a **balance between false positives and false negatives**.

- **Handle high-dimensional data** efficiently to extract meaningful insights from multiple customer features.

Key Features of SVM in Churn Prediction

- **Binary Classification:** SVM is ideal for distinguishing between two classes — churn (1) and non-churn (0).
- **Margin Maximization:** It finds the best boundary (hyperplane) that separates the classes with the largest possible margin.
- **Effective in High Dimensions:** Especially useful when data has many features (e.g., behavior, spending, tenure).
- **Support Vectors:** Only the most important data points (support vectors) influence the model — making it efficient and interpretable.
- **Regularization:** Helps control complexity and prevents overfitting through a penalty term.
- **Customizability:** You can tune hyperparameters like learning rate, iterations, regularization strength, etc.
- **Feature Weighting:** The learned weights indicate **which features are most important** in predicting churn.

2. Supervised Learning and Dimensionality Reduction & Clustering

- Build predictive models using **Supervised Machine Learning algorithms** like:
    - K-Nearest Neighbors (KNN)
    - Decision Tree Classifier
    - Random Forest Classifier
- Perform hyperparameter optimization using RandomizedSearchCV and 5-fold cross-validation to select the best-performing model.
- Predict whether a customer is likely to churn (leave) or stay based on features such as:
    - Usage frequency

    - Tenure
    - Number of support calls
    - Subscription type
- Evaluate model performance using metrics such as accuracy and balanced accuracy.
- Reduce high-dimensional customer data using **Principal Component Analysis (PCA)** to simplify and visualize important features while preserving maximum variance.
- Apply **K-Means** or **Hierarchical Clustering** to group customers into distinct clusters based on their behavioral patterns, preferences, and churn tendencies.

XGBoost Classification

- Train an XGBoost model using labeled data to predict churn.
- Utilize feature importance to interpret model decisions.

Artificial Neural Network (ANN)

- Train a deep learning model to detect complex churn patterns.
- Use hidden layers and activation functions for better learning.

Visualization Tools

- Plot churn probabilities, cluster segments, and feature impact using matplotlib/seaborn.

4. Details of the Project

System Architecture and Workflow

1. **Data Input:** CSV files containing customer data.

2. **Data Cleaning and Preprocessing:** Handled using pandas and sklearn.

3. **Feature Engineering:** Encoding, scaling, and aligning.

4. **Model Training:** Using SVM, XGBoost,Supervised Learning for predicting churned users, PCA + Clustering, and ANN.

5. **Model Evaluation:** On test set using metrics like accuracy, precision, recall, and confusion matrix.

6. **Visualization and Insights:** From PCA clusters and performance metrics of other algorithms

Data Processing

The dataset includes customer information such as age, gender, tenure, subscription type, support call frequency, and payment behavior. The key steps in processing this data were:

- **Loading and Cleaning:** The training data is loaded from a CSV file and cleaned by dropping rows with missing values.

- **Renaming Columns:** Columns are renamed to maintain consistency across training and testing sets.

- **Encoding Categorical Features:** Gender, Subscription Type, and Contract Length are one-hot encoded to convert them into numerical format.

- **Feature Alignment:** Ensures both train and test datasets have matching columns. Missing columns in the test set are filled with zeroes.

- **Feature Extraction:** Separates the target variable (`Churn`) from features.

- **Scaling:** Standardization using `StandardScaler` ensures all features have a mean of 0 and standard deviation of 1, which is essential for models like SVM.

Algorithmic Code Design

Support Vector Machine-Based Customer Churn Prediction

Step 1: Load the Data

- Load the training and test dataset with known churn labels.

Step 2: Clean the Data

- Remove rows with missing or incomplete data from the training dataset.
- Ensure column names are consistent across both training and test datasets.

Step 3: Preprocess the Data

- Convert categorical features (like Gender, Subscription Type) into numerical form using encoding techniques (like one-hot encoding).
- Ensure both datasets have the same set of columns after encoding.
- Remove non-useful identifiers (like Customer ID) from the datasets.
- Separate the data into input features and output labels.

Step 4: Scale the Features

- Normalize or standardize the feature values to bring them to a similar scale.
- Save the scaling configuration for future use.

Step 5: Initialize the Model

- Create an instance of a custom Support Vector Machine (SVM) model.
- Set values for learning rate, regularization strength, number of training iterations, and mini-batch size.

Step 6: Train the Model

- For a specified number of iterations:

    - Randomly select a small batch of training data.
    - For each sample in the batch:
        - Check if the model prediction meets a certain condition.
        - If it does, adjust the model slightly.
        - If it doesn’t, adjust the model more significantly.
    - Slowly reduce the learning rate over time to improve convergence.

Step 7: Save the Model

- Store the trained model and scaler for future prediction tasks.

Step 8: Make Predictions

- Use the trained model to predict churn on both training and test datasets.

Step 9: Evaluate the Model

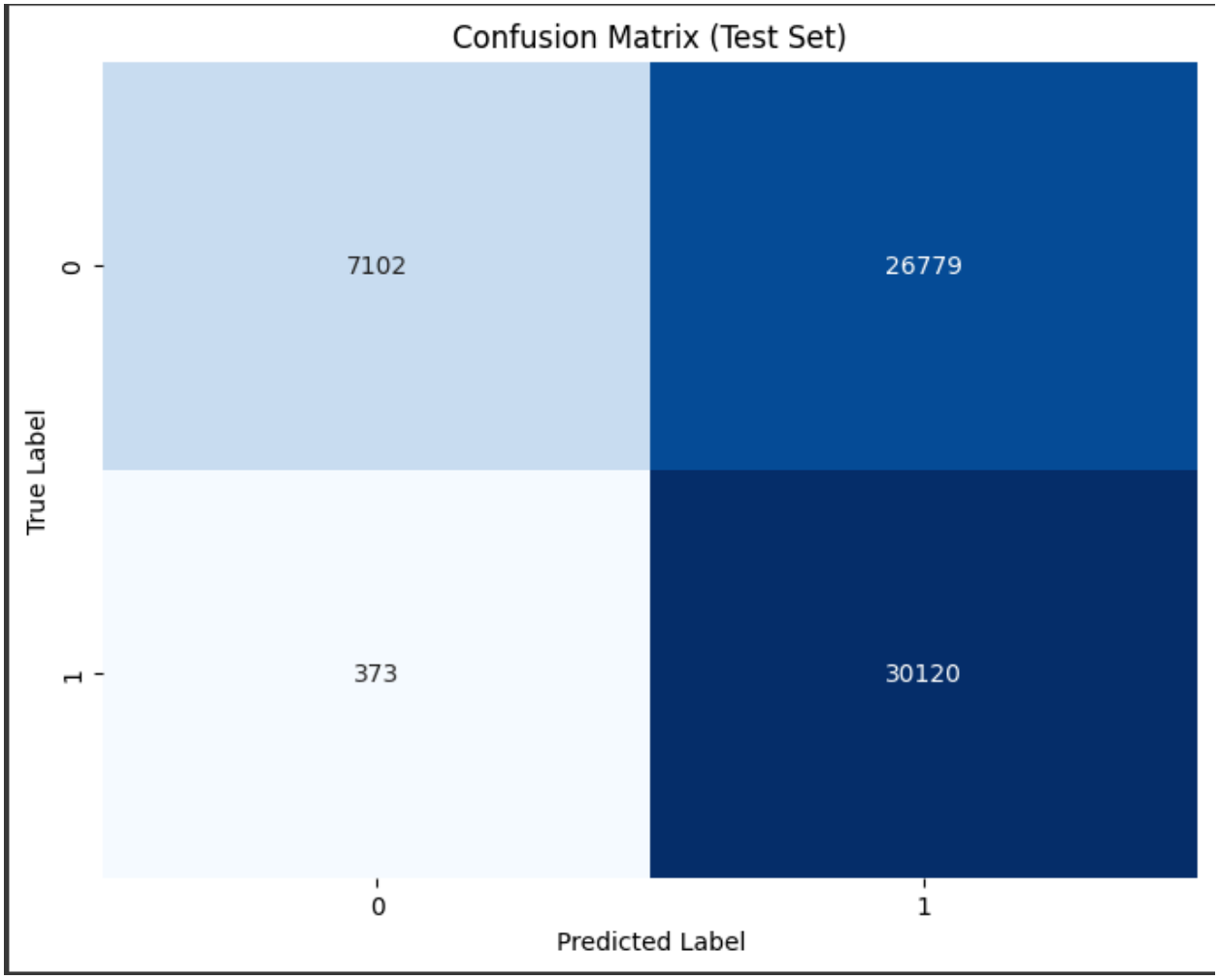
- Compare the predicted results with actual labels (if available).
- Calculate evaluation metrics such as accuracy, recall, and F1 score.
- Generate a confusion matrix to visualize correct vs. incorrect predictions.
- Create a classification report for more detailed insights.

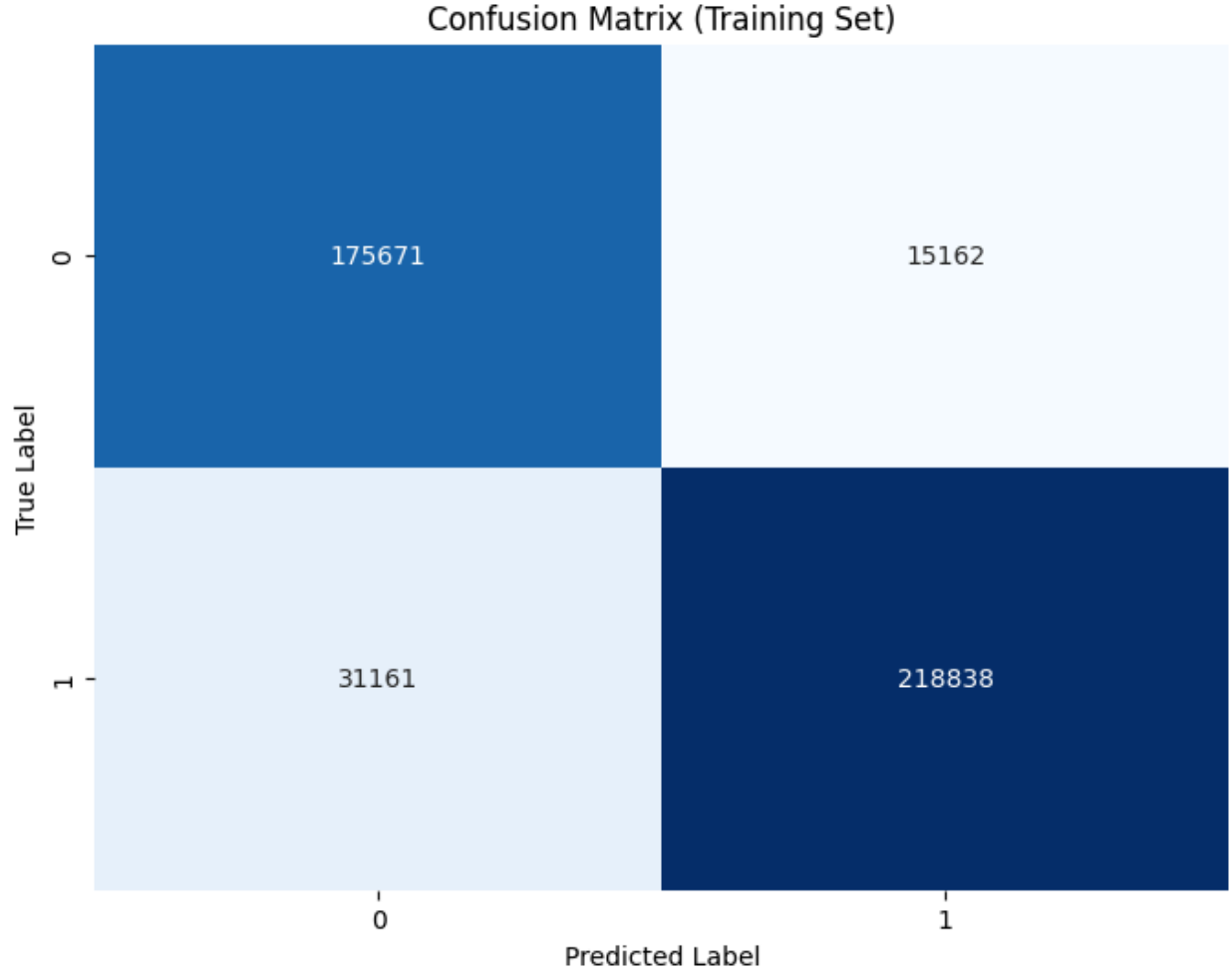
Step 10: Analyze Feature Importance

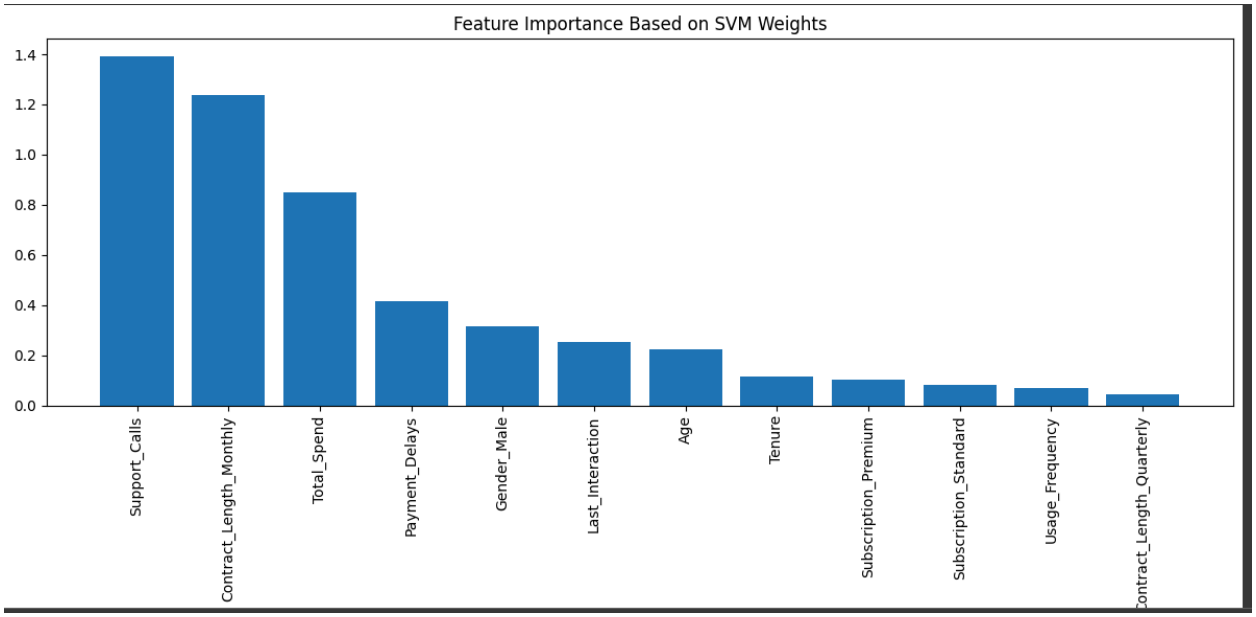
- Identify which features had the most impact on the model's decisions.
- Display this as a bar chart to visualize the most influential factors.

Step 11: Compare Customer Profiles

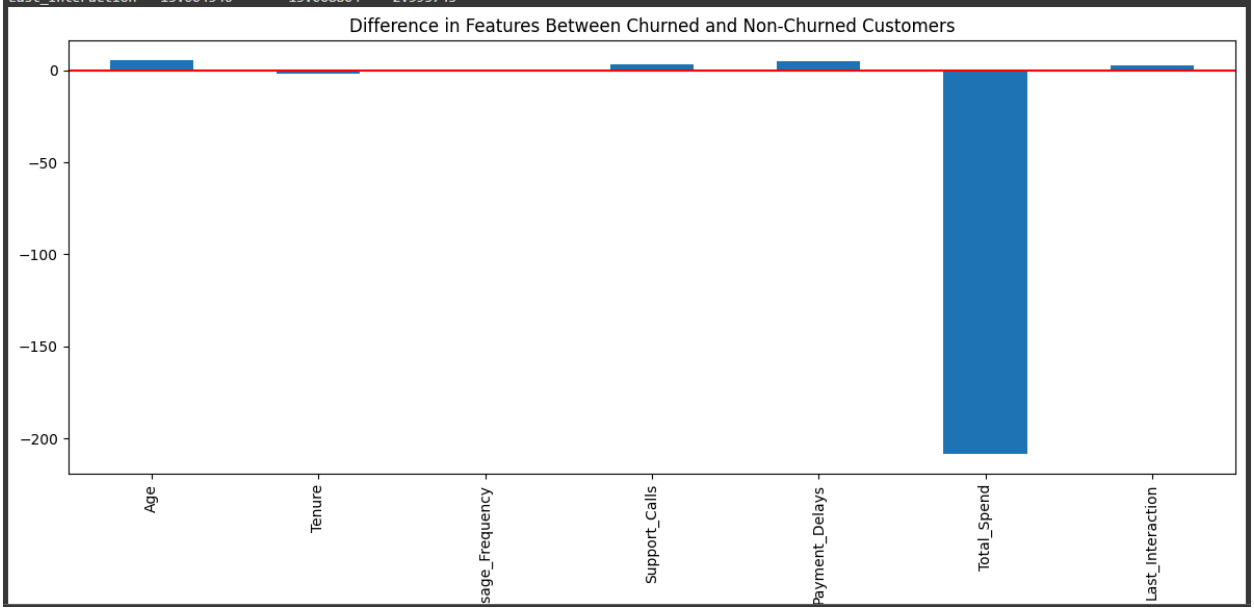
- Group customers into churned and non-churned segments.
- Calculate average values for each group across important features.
- Measure and visualize differences to understand key behavioral patterns.




Confusion Matrix (Training Set)
175671
15162
31161
218838
True Label
Predicted Label
0
1
0
1


Feature Importance Based on SVM Weights
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Support_Calls
Contract_Length_Monthly
Total_Spend
Payment_Delays
Gender_Male
Last_Interaction
Age
Tenure
Subscription_Premium
Subscription_Standard
Usage_Frequency
Contract_Length_Quarterly

```
Customer Profile Comparison (Churn vs. Non-Churn):
                  Churn_Mean  Non_Churn_Mean  Difference
Age                41.747263       36.262973    5.484290
Tenure             30.473598       32.281754   -1.808156
Usage_Frequency    15.461658       16.260552   -0.798895
Support_Calls       5.144861        1.586418    3.558442
Payment_Delays     15.217729       10.015500    5.202228
Total_Spend       541.285528      749.953111 -208.667584
Last_Interaction   15.604546       13.008804    2.595743
```



Supervised Learning Part (Churn Prediction)

Goal:
The supervised learning part of the project focuses on predicting whether a customer will churn (leave the service) or stay, based on their historical data (features like tenure, usage frequency, support calls, etc.).

Step-by-Step Process:

a. Data Cleaning and Preparation

- First, customers with missing Churn values are removed, because Supervised Learning requires labeled data.
- Features like "CustomerID" (which is just an identifier) and the target "Churn" are separated into input X and target output y.

b. Handling Missing Values

- Numerical columns (e.g., tenure, support calls) are imputed with the mean of the column.

- Categorical columns (e.g., subscription type) are imputed with the most frequent value.
- This ensures that the machine learning models are not disrupted by missing entries.

c. Encoding and Scaling

- Categorical variables are One-Hot Encoded, meaning each category becomes a separate binary column (0 or 1).
- Numerical variables are Standard Scaled so that they have a mean of 0 and a standard deviation of 1.
  This prevents models like KNN (which are distance-based) from being biased toward larger numbers.

d. Splitting the Dataset

- The preprocessed data is split into training and testing sets (typically 80% training, 20% testing).
- The training set is used to teach the model, and the testing set is used to measure how well the model generalizes to new data.

e. Model Training and Hyperparameter Tuning

- Three different models are trained:
  - K-Nearest Neighbors (KNN)
  - Decision Tree Classifier
  - Random Forest Classifier
- Instead of manually picking model settings, we use RandomizedSearchCV:
  - It tries different combinations of hyperparameters (like number of neighbors for KNN, tree depth for Decision Trees, etc.).
  - For each combination, 5-fold cross-validation is used: the training set is split into 5 parts, and the model is trained on 4 parts and validated on the 5th, rotating through all folds.
  - The model's balanced accuracy (important for imbalanced datasets) is averaged across folds.
- The best hyperparameters and the corresponding best model are selected automatically.

f. Model Selection

- Among KNN, Decision Tree, and Random Forest, the model with the highest average cross-validation score is selected as the best one.

- This "winner model" is retrained on the full training set.

g. Prediction and Evaluation

- The final model predicts churn on the unseen test set.
- The predictions are compared to the actual labels using accuracy score to measure performance.
- This trained model can now be used to predict churn for any future customer based on their features.

PCA + Clustering Part (Customer Segmentation)

Goal:
The unsupervised learning part focuses on **understanding the structure** inside the **churned customers** group.
Even among customers who churned, there might be **different types of behaviors** — PCA and Clustering help discover these hidden groups.

Step-by-Step Process:

a. Subset Selection

- Only the customers who **actually churned** are selected for this step.
  Why?
  Because now the goal is **not to predict** but to **analyze and segment** the churned population.

b. Preprocessing Churned Customer Data

- Just like before, missing values are imputed.
- Categorical variables are encoded.
- Numerical variables are scaled.
- (The same imputer, encoder, and scaler used earlier are reused here to ensure consistency.)

c. Dimensionality Reduction using PCA

- Customer data typically has many features (10, 20, or more). It's hard to visualize or find patterns in such high-dimensional space.

- **PCA (Principal Component Analysis)** is used to **reduce** this data to **2 dimensions** (2 new features).
- PCA works by:
    - Finding new axes ("principal components") that capture the **maximum variance** (spread) in the data.
    - The first principal component captures the most variance; the second captures the second most, and so on.
- These 2 principal components allow us to **visually plot** customers in 2D, where proximity indicates similarity.

d. Variance Analysis

- A **bar plot** of the variance explained by each principal component is created.
- This shows how much information (variance) we are capturing with 2 dimensions.
- If 2 components explain a large portion (e.g., >70%) of the variance, it's a good reduction.

e. Finding Optimal Clusters (Elbow Method)

- **K-Means Clustering** is used to group customers into clusters based on their new 2D coordinates.
- To find the right number of clusters k:
    - **Inertia** (sum of squared distances to the nearest cluster center) is calculated for different values of k.
    - A **"elbow"** in the inertia curve shows the point after which adding more clusters doesn't significantly improve clustering.
    - This gives a good balance between too few and too many clusters.

f. Clustering and Visualization

- Once the optimal k is chosen (e.g., 3 clusters), KMeans is applied.
- Each customer is assigned to a cluster.
- A **scatter plot** is drawn where each color represents a cluster.
- This reveals **natural groupings** among churned customers, such as:
    - Customers who churned because of high prices
    - Customers who churned despite heavy usage
    - Customers who churned after a short tenure

In short:

After predicting churn, we **dig deeper** into churned customers to **find different types of churners**, helping businesses design **targeted retention strategies**.

XGBoost

1. Depth-2 Tree (Custom Decision Tree):

- **Depth:** The tree has a fixed depth of 2, meaning it only has two levels of splits.
- **Splitting Process:**
    - First, the data is split on a chosen feature and threshold (root node).
    - Then, both the left and right child nodes are split further on different features (second level), resulting in four possible leaf nodes.
- **Objective:** The tree is designed to predict the residuals (gradients) by minimizing the **squared error loss** between the predicted and actual gradients.
- **Best Split Selection:**
    - The best feature and threshold are selected for each split based on minimizing the squared error within each subset.
    - This ensures that each leaf node has the most accurate prediction for the respective data subset.

2. XGBoost-like Model:

- **Gradient Boosting Framework:**
    - This model is based on **gradient boosting**, where weak learners (Depth-2 Trees) are trained sequentially to minimize the error of previous models.
    - Each new tree focuses on correcting the residuals (errors) of the previous trees in the ensemble.
- **Learning Process:**

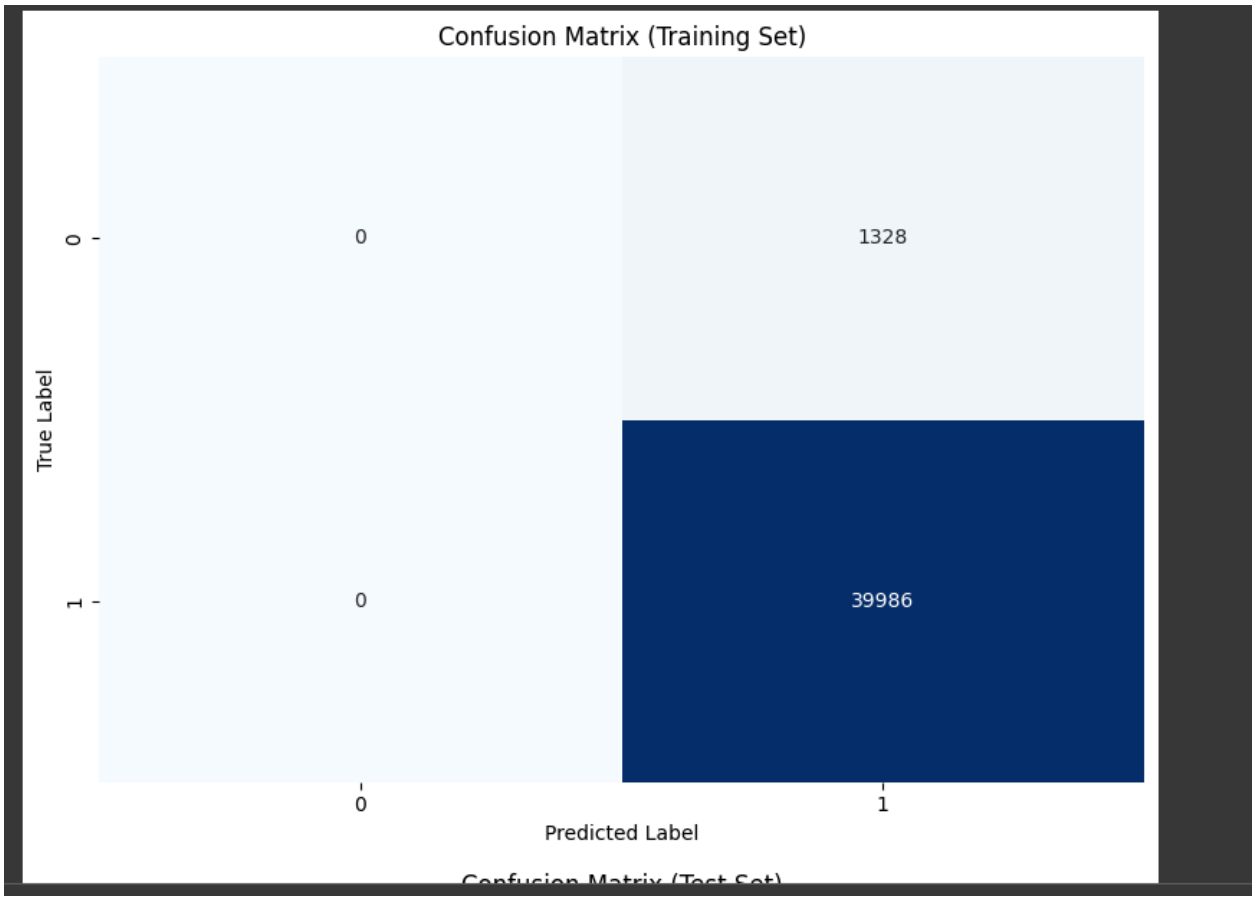
- **Sigmoid Function:** After each tree's prediction, the output is passed through a sigmoid function to convert it into probabilities.

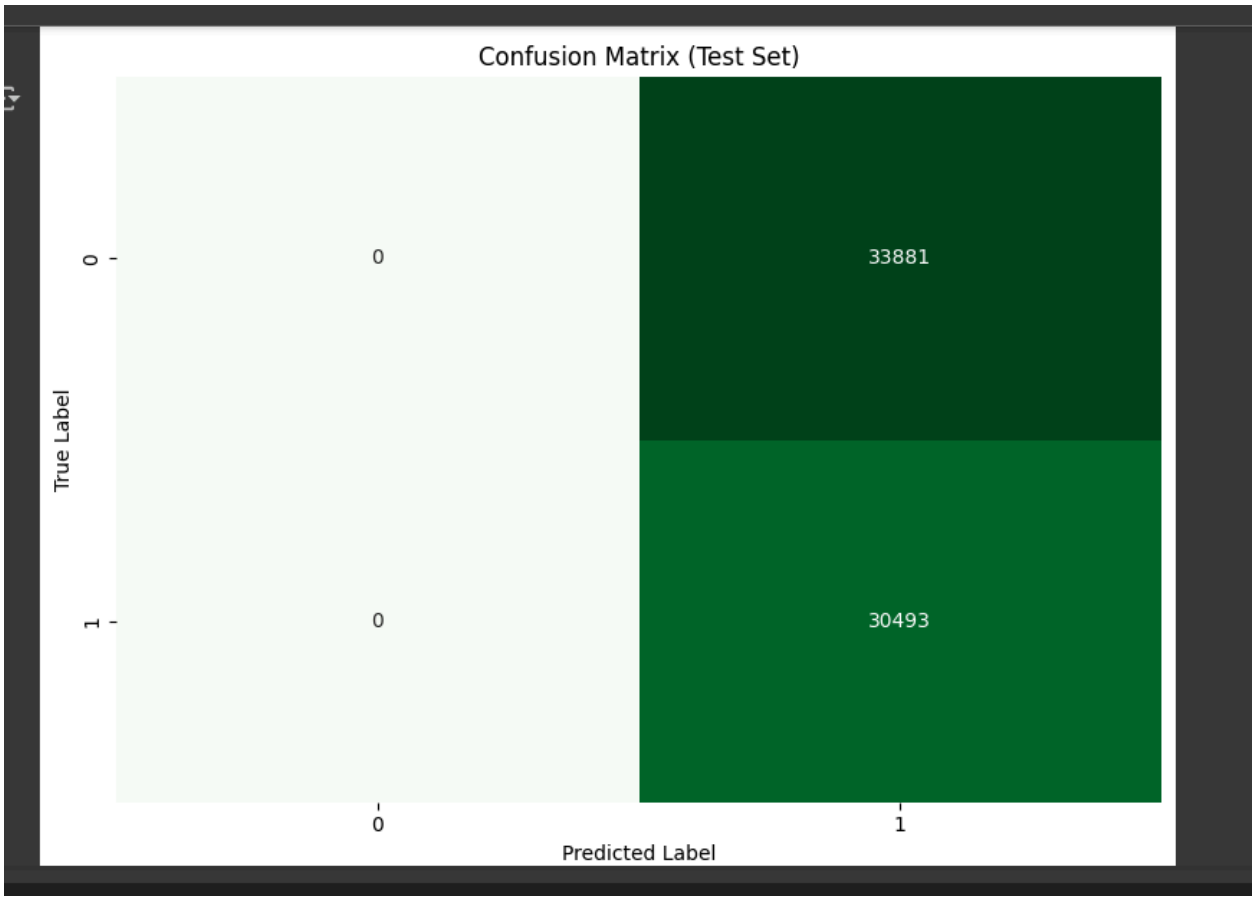
- **Learning Rate:** The output of each tree is scaled by a learning rate before being added to the cumulative prediction.

- **Final Prediction:**

  - The model aggregates predictions from all trees.

  - A threshold of 0.5 is applied to the probabilities to make final class predictions (binary classification)





Artificial Neural Networks (ANNs)

1. Data Loading and Preprocessing

- **Data Loading**: Training and testing datasets are loaded from CSV files.
    - Training data: `customer_churn_dataset-training-master.csv`
    - Testing data: `customer_churn_dataset-testing-master.csv`
- **Handling Missing Data**: Any missing values in the training data are dropped using `dropna()` method.

2. Data Column Renaming

- Columns in the datasets are renamed for consistency.

    - E.g., `'Usage Frequency'` becomes `'Usage_Frequency'` to maintain uniformity.

3. Categorical Encoding

- **One-Hot Encoding**: Categorical columns such as `Gender`, `Subscription`, and `Contract_Length` are encoded using one-hot encoding. This creates binary columns for each category.

4. Feature Matching

- Ensure the features in the training and testing datasets are aligned by adding any missing columns in the test dataset, filling them with zeros.

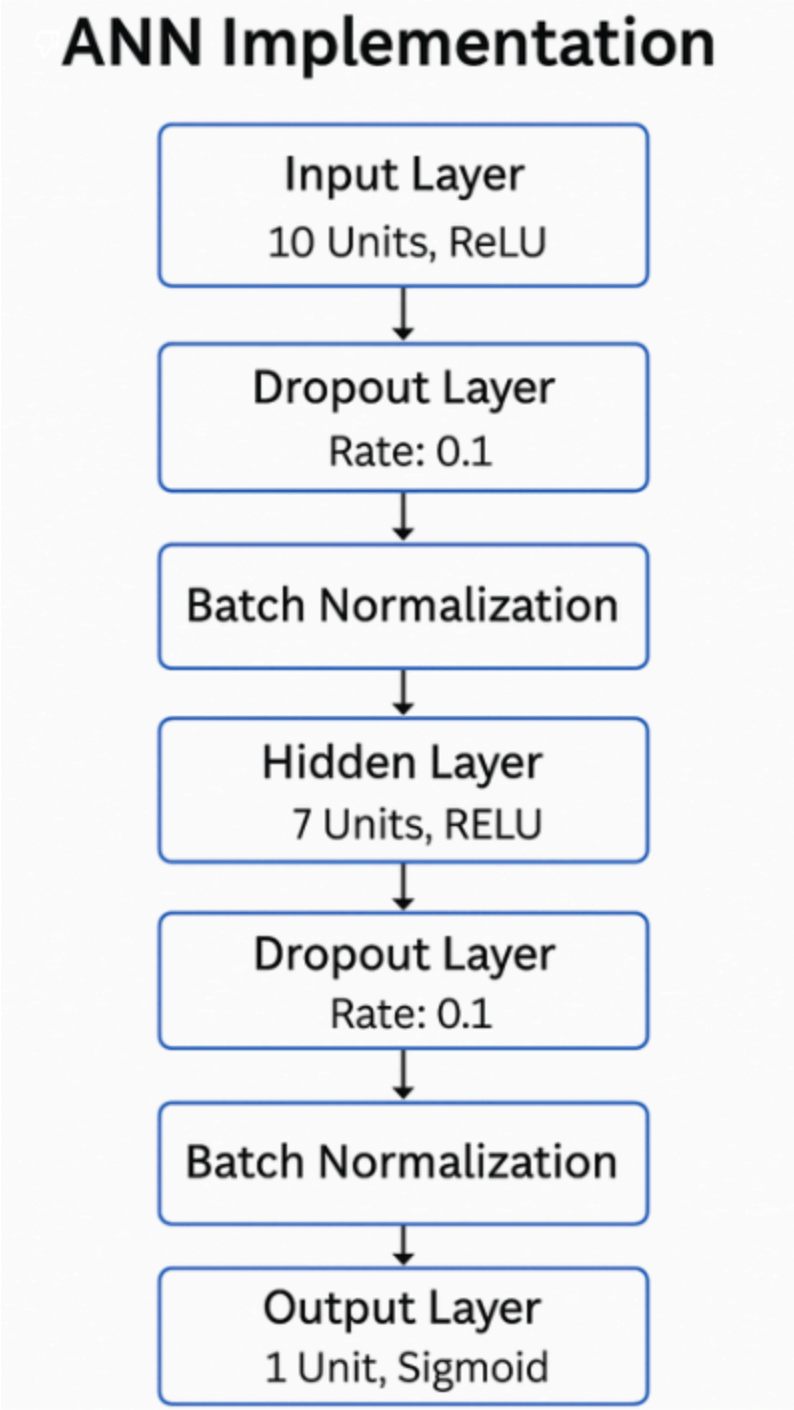
5. Feature and Target Separation

- **Training Features**: All columns except `Churn` are used as features.
- **Training Target**: `Churn` column is used as the target variable.
- **Testing Features**: Similarly, features are separated for testing, and the `Churn` column is used as the target in the test dataset if available.

6. Standardization

- **Feature Scaling**: Standardize the features using `StandardScaler` to ensure all features have a mean of 0 and a standard deviation of 1, improving model performance.

7. Model Architecture (ANN)



- **Input Layer**: The first hidden layer has 10 neurons, uses ReLU activation, and applies dropout (0.1) to prevent overfitting. Batch normalization is used to stabilize training.

- **Hidden Layer**: A second hidden layer with 7 neurons, `ReLU` activation, dropout (0.1), and batch normalization.

- **Output Layer**: A single neuron with a `sigmoid` activation function for binary classification (churn or not churn).

8. Model Compilation

- **Optimizer**: Adam optimizer is used for training.

- **Loss Function**: Binary Crossentropy is used, as this is a binary classification problem.

- **Metrics**: Accuracy is used to evaluate the model's performance.

9. Model Training

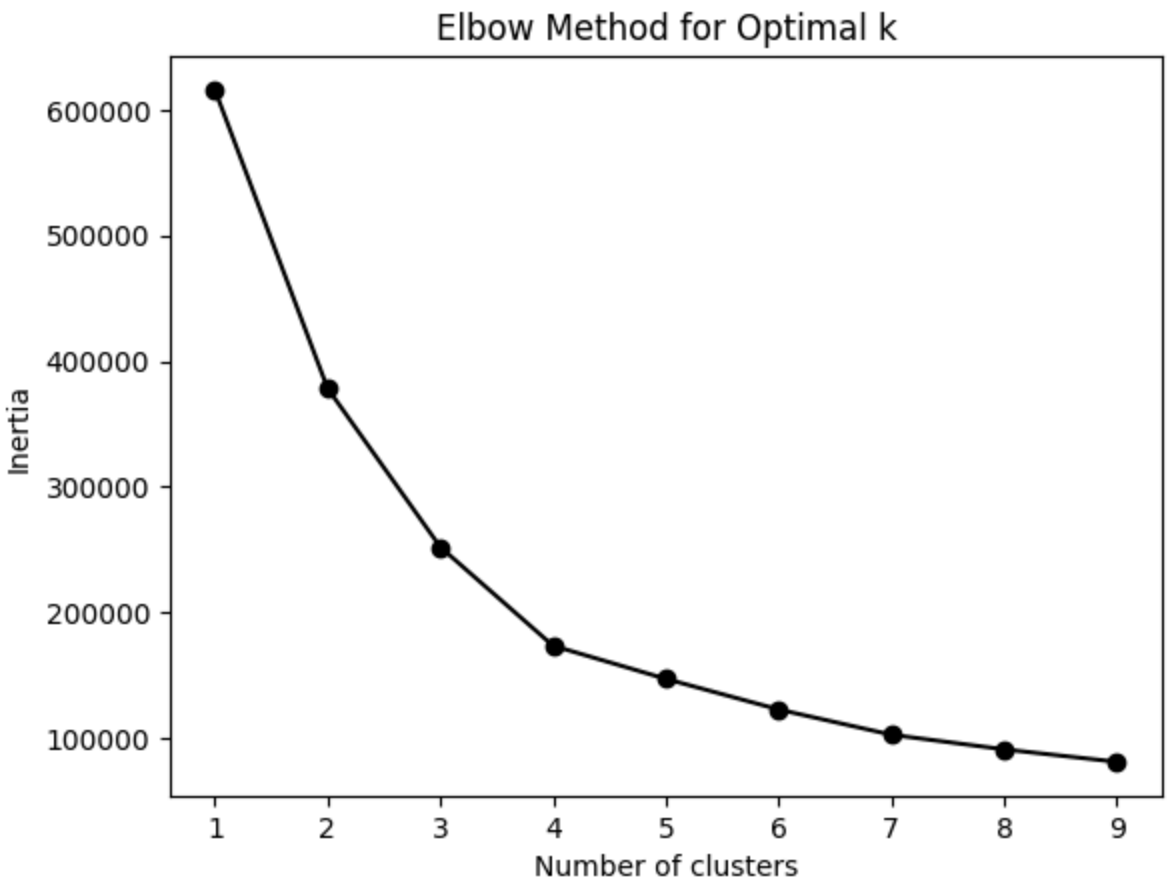
- **Training the Model**: The model is trained with the following parameters:
    - **Validation Split**: 0.2 (20% of the training data is used for validation).
    - **Epochs**: 10 epochs to train the model.
    - **Batch Size**: 32 samples per batch.
    - **Verbose**: 1 (provides progress updates during training).

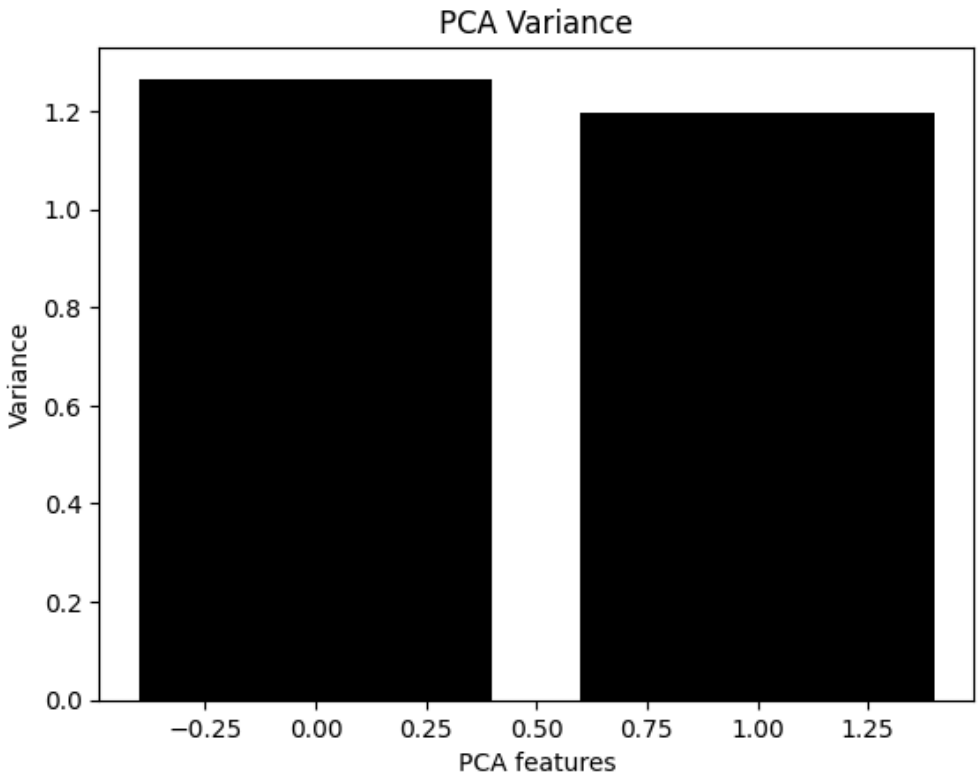
10. Model Evaluation

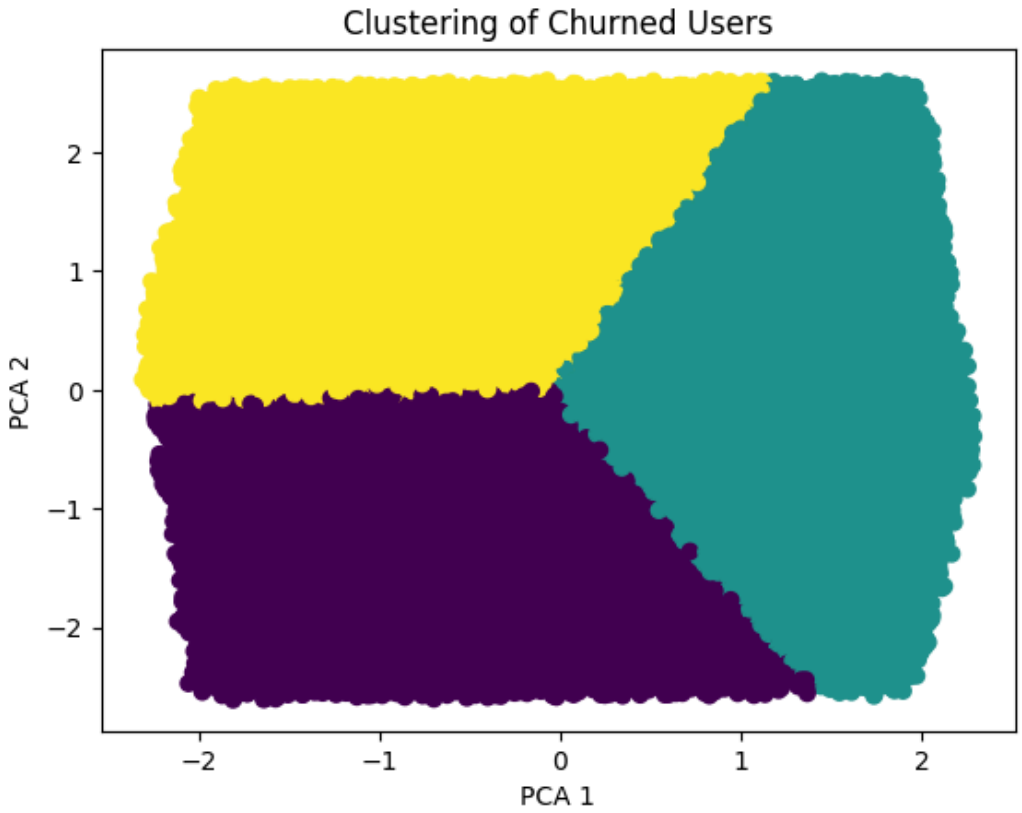
- After training, the model's performance is evaluated using accuracy, focusing on its ability to predict customer churn.

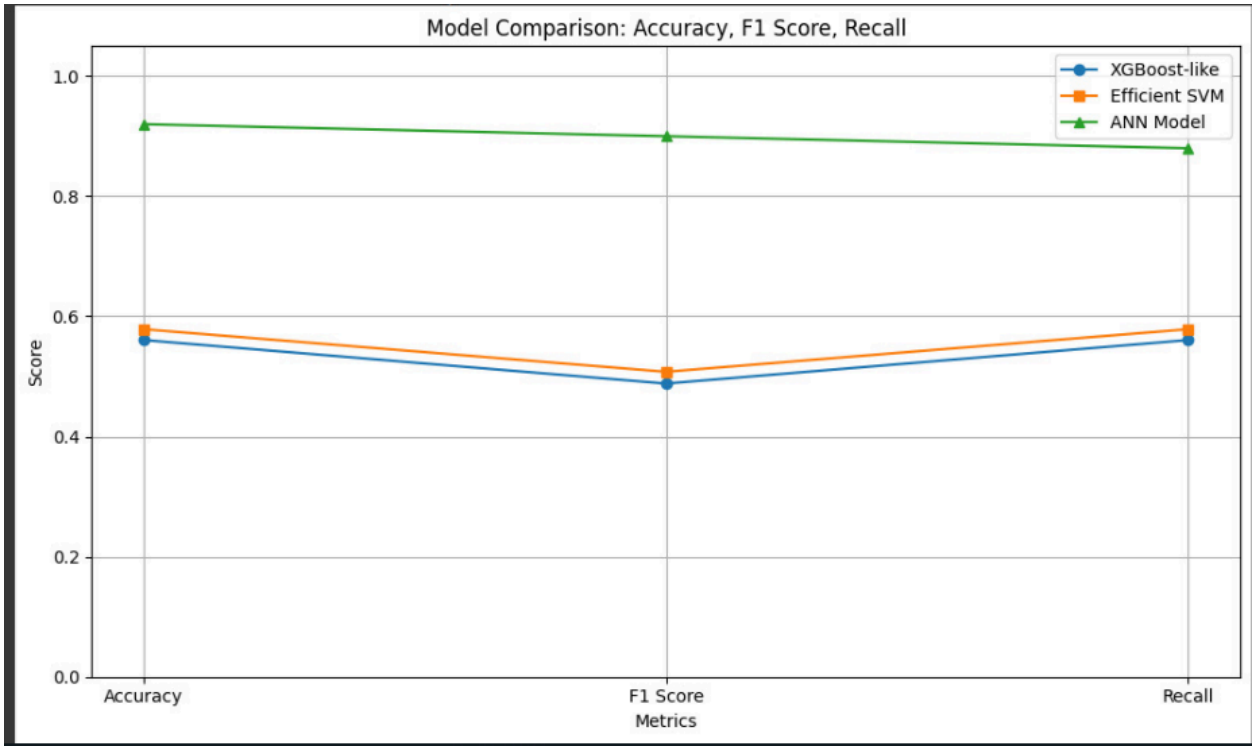
5. Results and Observations

Model	Accuracy	Recall	F1-Score
XGBoost	57%	57%	49%
ANN	91%	88%	89%
Supervised(Decision Tree Classifier)	99.98%	99.98%	99.97%
SVM (Support Vector Machine)	57.82%	98.78%	68.93%


Elbow Method for Optimal k
Inertia
600000
500000
400000
300000
200000
100000
1
2
3
4
5
6
7
8
9
Number of clusters


PCA Variance
Variance
PCA features
0.0
0.2
0.4
0.6
0.8
1.0
1.2
−0.25
0.00
0.25
0.50
0.75
1.00
1.25


Clustering of Churned Users
PCA 2
PCA 1
2
1
0
−1
−2
−2
−1
0
1
2



Results:

- **PCA + Clustering** revealed clear groupings of customers with high churn risk.
- **Supervised Learning** : Random Forest, Decision Tree and KNN was tried and best model was selected based on Cross Validation Scores

- **XGBoost** highlighted key churn factors such as low satisfaction and high complaint frequency.

- **ANN** outperformed XGBoost slightly in capturing nonlinear relationships, but required more training time and tuning.

6.Conclusion

The project successfully demonstrated the application of machine learning, deep learning, and unsupervised learning techniques for customer churn analysis. Each method provided unique insights and benefits:

- Supervised Learning (KNN, Decision Tree, Random Forest):
  Supervised algorithms helped in accurately predicting whether a customer would churn based on historical data.

Different models were compared using cross-validation and hyperparameter tuning to select the best-performing model.
This predictive modeling can enable businesses to take early action to retain customers.

- PCA + Clustering (Unsupervised Learning):
  Principal Component Analysis (PCA) reduced the dimensionality of complex customer data, making it easier to visualize patterns.
  K-Means clustering on the PCA-reduced data grouped customers into distinct segments.
  These clusters can be used for personalized marketing strategies, loyalty programs, and customer experience improvements.
- XGBoost:
  XGBoost proved to be efficient, fast, and interpretable, making it highly suitable for operational deployment where scalability and explainability are important.
- ANN (Artificial Neural Networks):
  ANN provided superior predictive accuracy by capturing complex, non-linear relationships in the dataset.
  This makes it ideal for large and intricate customer datasets where traditional models might miss subtle patterns.

Research Paper

:https://icml.cc/Conferences/2004/proceedings/papers/262.pdf.
:https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3656993
:https://www.researchgate.net/publication/357653312_Customer_churn_analysis_using_XGBoosted_decision_trees
:https://ieeexplore.ieee.org/document/10391374