Predicting Customer Churn:

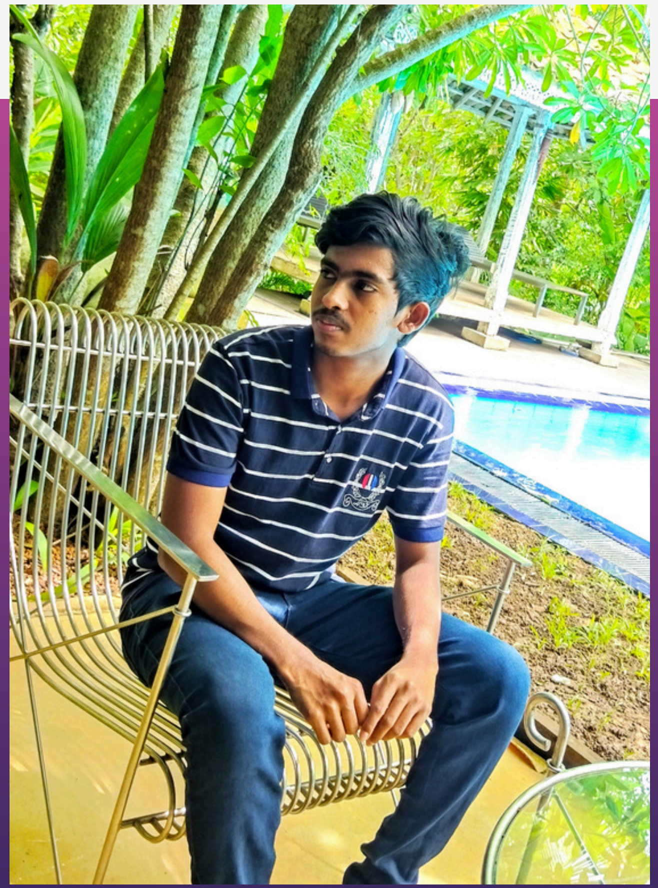
A MACHINE LEARNING APPROACH

Team 4

Meet THE TEAM

Samuditha Saradindu

Dilan Madhusankha



Nipun Lakshitha

Dileka Ratnayaka

INTRODUCTION

Project Overview:
Classification of Bank Customer Churn status using Machine Learning

Objectives:

- Develop predictive models for identifying and classify customers likely to churn.
- Evaluate model performance and provide actionable insights to banks.

IMPORTANCE OF CHURN PREDICTION & CLASSIFICATION

1. **Retention Strategy**: Identifying and retaining valuable customers is essential for sustainable business growth.
2. **Cost Reduction**: Acquiring new customers is more expensive than retaining existing ones. Predicting churn helps allocate resources efficiently.
3. **Customer Satisfaction**: Proactively addressing churn drivers improves overall customer satisfaction and loyalty.
4. **Competitive Advantage**: Effective churn management provides a competitive edge by maintaining a loyal customer base.

Dataset DESCRIPTION

Feature Variables:

- credit_score: The credit score of the customer.
- country: The country where the customer resides.
- gender: The gender of the customer.
- age: The age of the customer.
- tenure: Number of years the customer has been holding an account with the bank.
- Account balance: The balance in the customer's account.
- products_number: The number of products the customer has with the bank.
- credit_card: Indicator variable (binary) indicating if the customer has a credit card.
- active_member: Indicator variable (binary) indicating if the customer is an active member of the bank.
- estimated_salary: The estimated salary of the customer.

Target Variable:

- churn: Churn status of the customer. This is the variable that indicates whether the customer has churned (left the bank) or not.

Feature ENGINEERING

Customer ID Removal

We dropped the Customer ID column as it serves as a unique identifier with no predictive value for churn

One-Hot Encoding

Categorical variables such as country and gender were one-hot encoded to transform them into binary representations suitable for modeling

Handling Missing Values

Missing values in numerical columns were filled using mean, to ensure completeness of the dataset.

Numerical Feature Scaling

numerical features like credit score, age, tenure, account balance, and estimated salary scaled to a common range to improve model performance

USED MACHINE LEARNING MODELS

01
K-nearest neighbor (KNN)

02
Naive Bayes

03
Logistic regression

04
Support vector machine

05
Descion Tree

06
Random Forest

07
Deep Neural Networks (DNN)

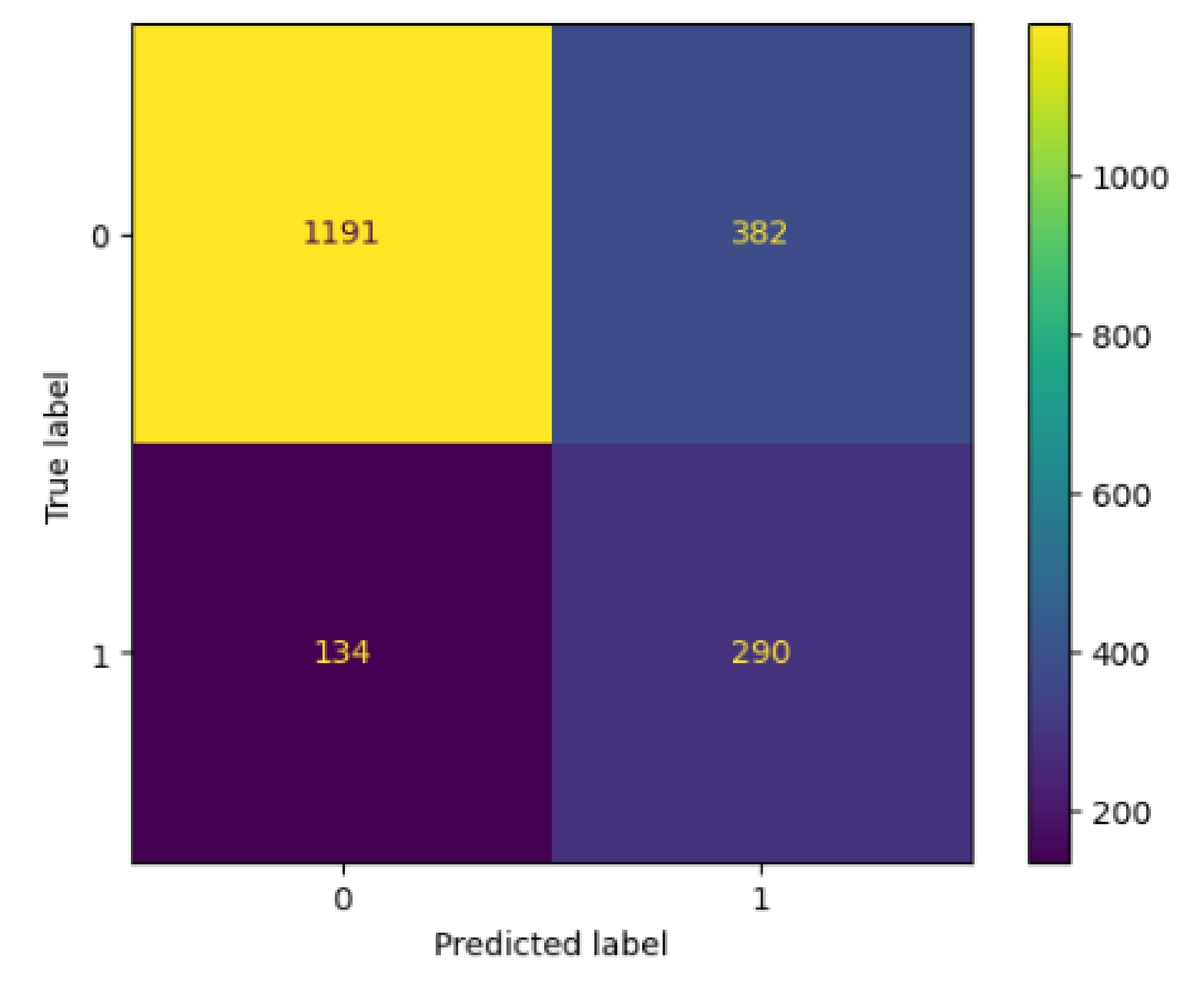
08
Boosting algorithms

01. K-nearest neighbor (KNN)

- KNN is a non parametric algorithm and it classifies or predicts based on the majority class of the k-nearest data points in the feature space.
- k-NN identifies the k nearest neighbors of a given data point based on a chosen distance metric
- Since k-NN is a non-parametric algorithm, it makes minimal assumptions about the underlying data distribution. This flexibility allows it to capture complex relationships between features and the target variable

01. KNN Results

Undersampling gives the best result



Classification Report:

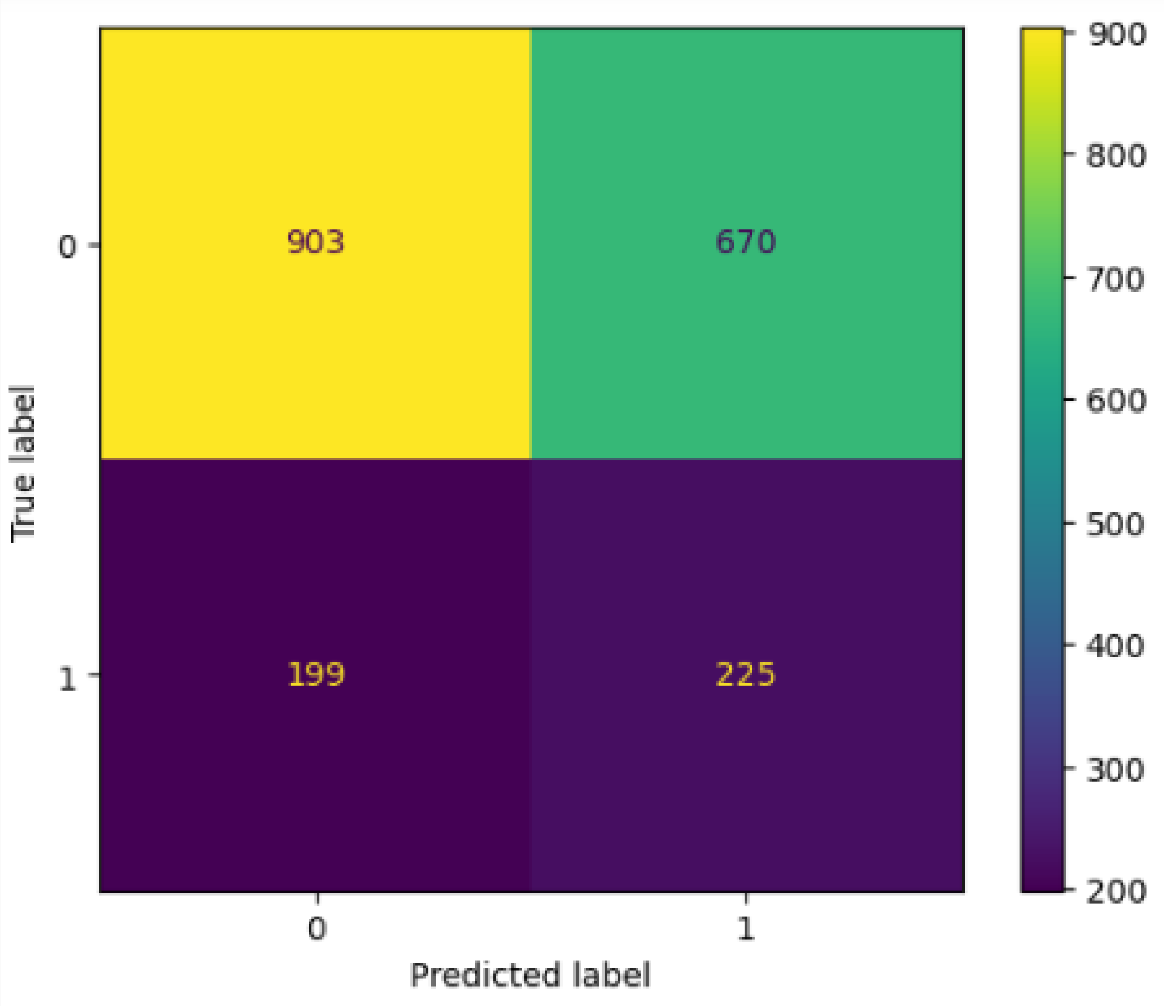
	precision	recall	f1-score	support
0	0.90	0.76	0.82	1573
1	0.43	0.68	0.53	424
accuracy			0.74	1997
macro avg	0.67	0.72	0.68	1997
weighted avg	0.80	0.74	0.76	1997

02. Naive Bayes

- Naive Bayes is a probabilistic algorithm based on Bayes' theorem. It calculates the probability of a certain event (in this case, churn) given the presence of certain features.
- Naive Bayes works by calculating the probabilities of each class given the observed features using Bayes' theorem and the assumption that features are conditionally independent given the class label, then selecting the class with the highest probability as the prediction.
- Naive Bayes is computationally efficient and scales well with the size of the dataset. It's particularly suitable for large datasets with many features.

02. Naive Bayes Results

Undersampling and oversampling gives the same result



Classification Report:

	precision	recall	f1-score	support
0	0.82	0.57	0.68	1573
1	0.25	0.53	0.34	424
accuracy			0.56	1997
macro avg	0.54	0.55	0.51	1997
weighted avg	0.70	0.56	0.60	1997

03. Logistic Regression

- A linear model commonly used for binary classification tasks like churn prediction.
- It models the probability of a binary outcome (churn or no churn) based on one or more predictor variables.
- known for its simplicity, interpretability, and efficiency, making it a popular choice for classification tasks.

```
Best Hyperparameters: {'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
Accuracy with Best Model: 0.8126873126873126
Confusion Matrix with Best Model:
[[1554   75]
 [ 300   73]]
Classification Report with Best Model:
              precision    recall  f1-score   support

           0       0.84      0.95      0.89      1629
           1       0.49      0.20      0.28       373

    accuracy                           0.81      2002
   macro avg       0.67      0.57      0.59      2002
weighted avg       0.77      0.81      0.78      2002
```

04. Support Vector Machines (SVM)

- A versatile supervised learning algorithm capable of performing both linear and non-linear classification tasks.

- It works by finding the optimal hyperplane that separates classes in a high-dimensional space, maximizing the margin between data points.

- SVM is robust to overfitting and performs well in high-dimensional spaces, making it suitable for complex classification tasks like churn prediction.

```
Best Hyperparameters: {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}
Best Score: 0.8541979231730169

Accuracy with Best Model: 0.8656343656343657
Confusion Matrix with Best Model:
[[1587   42]
 [ 227  146]]
Classification Report with Best Model:
              precision    recall  f1-score   support

           0       0.87      0.97      0.92      1629
           1       0.78      0.39      0.52       373

    accuracy                           0.87      2002
   macro avg       0.83      0.68      0.72      2002
weighted avg       0.86      0.87      0.85      2002
```

05. Decision Tree

What is Decision Tree?

- Supervised learning algorithm used for classification and regression.
- Represents decisions in a tree-like structure based on feature values.

Why Decision Tree?

- Easy to interpret and visualize.
- Handles both numerical and categorical data efficiently.

What Happens in Decision Tree?

- Splits data into subsets based on feature values to maximize information gain.
- Continues recursively until a stopping condition is met, forming decision nodes and leaf nodes.

05. Decision Tree Results

```
Best Parameters: {'max_depth': 5, 'max_features': None, 'min_samples_leaf': 2, 'min_samples_split': 2}
Accuracy: 0.84

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.96      0.91      1556
           1       0.75      0.44      0.55       440

    accuracy                           0.84      1996
   macro avg       0.80      0.70      0.73      1996
weighted avg       0.83      0.84      0.83      1996


Confusion Matrix:
[[1492   64]
 [ 247  193]]
```

Without Balancing the dataset

```
Best Parameters: {'max_depth': 7, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 5}
Accuracy: 0.84

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.98      0.90      1556
           1       0.81      0.33      0.47       440

    accuracy                           0.84      1996
   macro avg       0.83      0.65      0.68      1996
weighted avg       0.83      0.84      0.81      1996


Confusion Matrix:
[[1523   33]
 [ 296  144]]
```

Algorithmic Techniques

```
Best Parameters: {'max_depth': 10, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 5}
Accuracy: 0.81

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.87      0.88      1556
           1       0.56      0.59      0.57       440

    accuracy                           0.81      1996
   macro avg       0.72      0.73      0.73      1996
weighted avg       0.81      0.81      0.81      1996


Confusion Matrix:
[[1353  203]
 [ 181  259]]
```

Oversampling using SMOTE
(Synthetic Minority Over-sampling Technique).

```
Best Parameters: {'max_depth': 5, 'max_features': None, 'min_samples_leaf': 8, 'min_samples_split': 2}
Accuracy: 0.75

Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.77      0.76       422
           1       0.75      0.73      0.74       392

    accuracy                           0.75       814
   macro avg       0.75      0.75      0.75       814
weighted avg       0.75      0.75      0.75       814


Confusion Matrix:
[[327  95]
 [106 286]]
```

Under sampling

06. Random Forest

What is Random Forest?

- Ensemble learning method that builds multiple decision trees and combines their predictions.
- Each tree is trained on a bootstrap sample of the data and a random subset of features.

Why Random Forest?

- Improves prediction accuracy and reduces overfitting compared to individual decision trees.
- Handles high-dimensional data and large datasets effectively.

What Happens in Random Forest?

- Constructs a forest of decision trees using bootstrapping and random feature selection.
- Combines predictions from individual trees through voting to make the final prediction.

06. Random Forest Results

```
Best Parameters: {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 50}
Accuracy: 0.85

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.96      0.91      1556
           1       0.77      0.47      0.58       440

    accuracy                           0.85      1996
   macro avg       0.82      0.71      0.74      1996
weighted avg       0.84      0.85      0.84      1996


Confusion Matrix:
[[1494   62]
 [ 235  205]]
```

07. Deep Neural Networks (DNN)

- DNNs, or Deep Neural Networks, are advanced models composed of interconnected layers that process data to make predictions, mimicking the human brain's structure.

- DNNs function by passing data through layers of interconnected nodes, refining information to extract patterns and features. Through iterative weight adjustments during training, DNNs learn to map input data to predictions, facilitating sophisticated decision-making.

- Deep neural networks are great for predicting & classifying bank customer churn because they handle complex data well and aren't easily overwhelmed by large amounts of information.

```
Epoch 48/50
251/251 [==============================] - 1s 4ms/step - loss: 0.5115 - accuracy: 0.7920 - val_loss: 0.4813 - val_accuracy:
0.8137
Epoch 49/50
251/251 [==============================] - 1s 4ms/step - loss: 0.5118 - accuracy: 0.7920 - val_loss: 0.4845 - val_accuracy:
0.8137
Epoch 50/50
251/251 [==============================] - 1s 4ms/step - loss: 0.5117 - accuracy: 0.7920 - val_loss: 0.4818 - val_accuracy:
0.8137
```

08. Boosting Algorithms

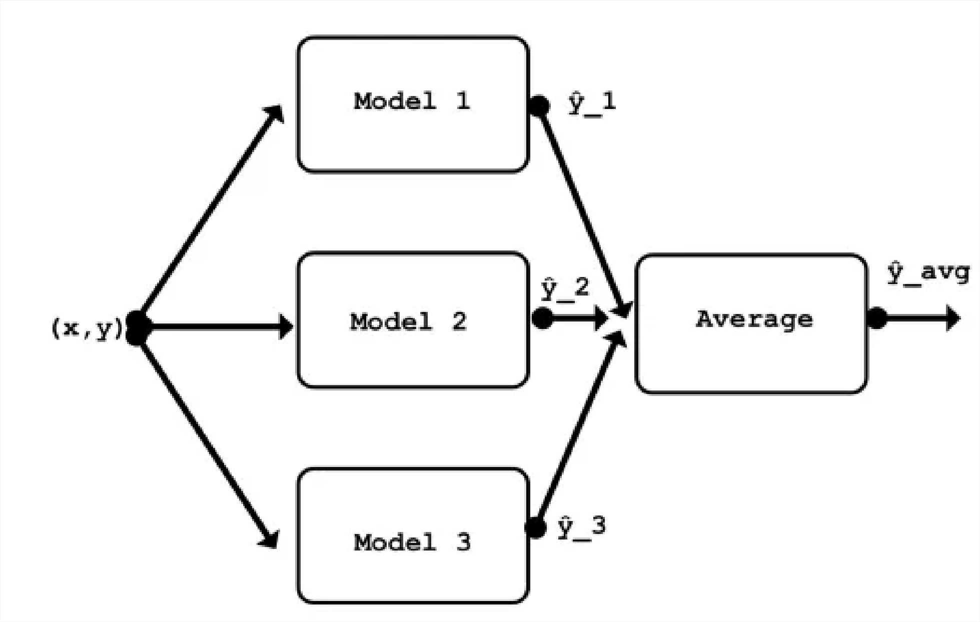
- A set of machine learning techniques that combine multiple weak models to create a strong predictive model.

- They sequentially train weak learners to correct errors made by previous models, leading to a strong ensemble model with improved predictive accuracy.

- Boosting algorithms excel in bank customer churn status classification by learning from past mistakes, identifying subtle churn indicators in complex banking data. Through iterative learning, they refine predictive models, enhancing accuracy in identifying potential churners.

```
Accuracy: 0.7737737737737738
Precision: 0.507337526205450 7
Recall: 0.5272331154684096
F1-score: 0.5170940170940171
Confusion Matrix:
[[1304  235]
 [ 217  242]]
```

Ensembling for Better Churn Prediction



KNN showed the best performance among all the individual models

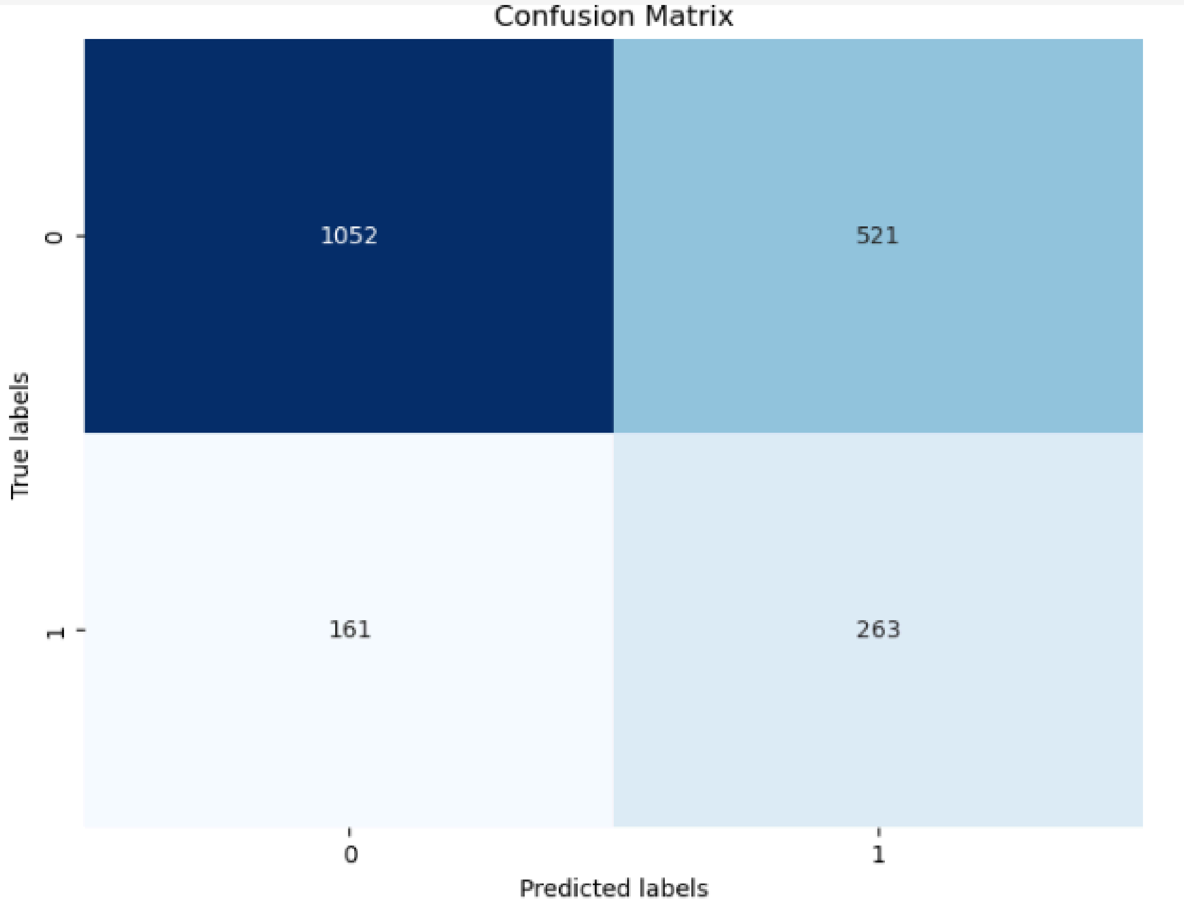
The model correctly identifies 68% of instances belonging to the positive class which represent the high risk churn customers

stacking and voting approaches were utilized to improve prediction accuracy

Magority

- each base classifier in the ensemble predicts a class label for a given sample, and the final prediction is determined by the most commonly chosen class label among the individual classifiers.

- The VotingClassifier from the scikit-learn library is utilized to combine the predictions of multiple individual classifiers into a single, aggregated prediction

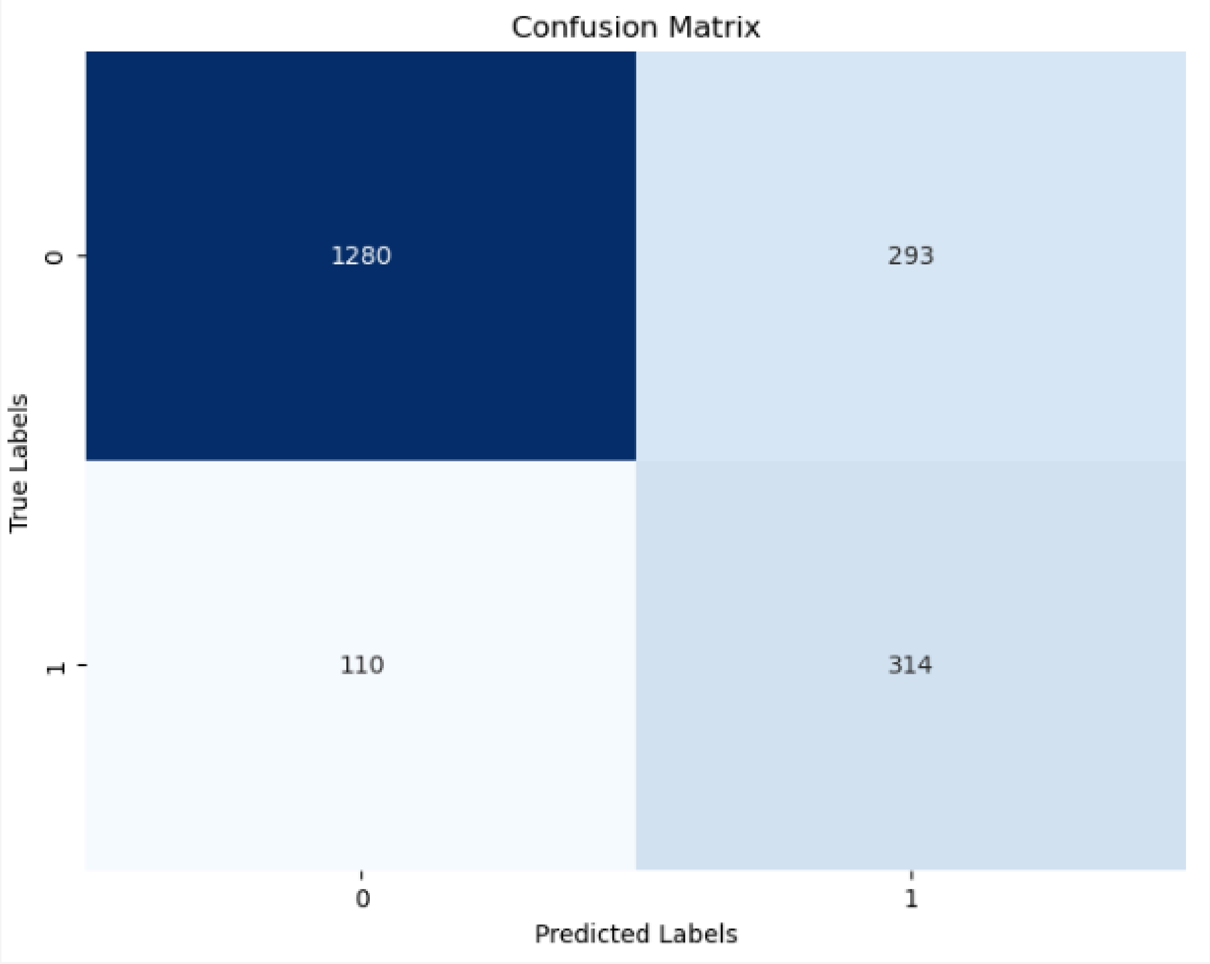



```
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.67      0.76      1573
           1       0.34      0.62      0.44       424

    accuracy                           0.66      1997
   macro avg       0.60      0.64      0.60      1997
weighted avg       0.75      0.66      0.69      1997
```

Stacking

- Stacking works by training several different models, then using the predictions from these models as input to a higher-level model, which learns how to best combine them to produce a final prediction.

- The StackingClassifier in scikit-learn combines predictions from multiple base classifiers by training a higher-level model to learn how to best combine them into a final prediction.



Classification Report:

	precision	recall	f1-score	support
0	0.92	0.81	0.86	1573
1	0.52	0.74	0.61	424
accuracy			0.80	1997
macro avg	0.72	0.78	0.74	1997
weighted avg	0.84	0.80	0.81	1997

Conclusion

After evaluating the performance of various individual models and ensemble techniques on the dataset, it was found that KNN yielded the best performance among the individual models, achieving an accuracy of 74% and recall of positive class of 0.68 . However, when employing ensemble techniques, stacking outperformed other methods, attaining an accuracy of 80% and recall of positive class of 0.74. It is important to highlight that accurately identifying instances of the positive class (high-risk class) is crucial, and stacking exhibited better recall and precision for this class compared to individual models

THANK YOU

THANK YOU