CUSTOMER CHURN PREDICTION USING MACHINE LEARNING

A Project Report
In the partial fulfilment of the award of the degree of

Bachelor of Technology

Under

Academy of Skill Development

ASD
Skills For Employment

Submitted by:

ANJAN KUMAR RAJAK

बिरला प्रौद्योगिकी संस्थान
Birla Institute of Technology
Mesra, Ranchi (India) - 835215

BIRLA INSTITUTE OF TECHNOLOGY, MESRA - RANCHI

CERTIFICATE FROM THE MENTOR

This is to certify that **ANJAN KUMAR RAJAK** has completed the project titled CUSTOMER CHURN PREDICTION USING MACHINE LEARNING under my supervision during the period from May to July which is in partial fulfilment of requirements for the award of the Bachelor of Technology and submitted to Department of Production and Industrial Engineering of BIRLA INSTITUTE OF TECHNOLGY , MESRA .

DATE: **Signature of the Mentor**

ACKNOWLEDGMENT

I take this opportunity to express my deep gratitude and sincerest thanks to my project mentor, **Mr.Joyjit Guha Biswas** for giving the most valuable suggestions, helpful guidance, and encouragement in the execution of this project work.

I would like to give a special mention to my colleagues. Last but not least I am grateful to all the faculty members of the **Academy of Skill Development** for their support.

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information of approval in any respect will be summarily rejected.)

1. Name of the Student: ANJAN KUMAR RAJAK

2. Title of the Project: **CUSTOMER CHURN PREDICTION USING MACHINE LEARNING**

3. Name and Address of the Guide: **MR. JOYJIT GUHA BISWAS SR.**
Sr. Subject Matter Expert & Technical Head (Python) Academy of Skill Development (An ISO 9001:2008 Certified) Module-132, SDF Building Salt Lake Sector-V, Kolkata - 700 091

4. Educational Qualification of the Guide: Ph.d* ☐ M.tech* ☐ B.E*/B.Tech * ☐ MCA* ☐ M.Sc* ☐

5. Working and Teaching experience of the Guide: **…….Years**
6. Software used in the Project:
   a. **Google collab**
   b. **Python**
   c. **Jupyter Notebook**

Signature of the Guide

Date:

Name: Mr. Joyjit Guha Biswas

Subject Matter Expert

Signature, Designation, Stamp of the Project Proposal Evaluator

APPROVED NOT APPROVED

SELF- CERTIFICATE

This is to certify that the dissertation/project proposal entitled "Customer Churn Prediction using Machine Learning" is done by me, under the guidance of Mr. Joyjit Guha Biswas. The matter embodied in this project work has not been submitted earlier for award of any certificate to the best of our knowledge and belief.

Name of the Students: ANJAN KUMAR RAJAK

Signature of the students:

CERTIFICATE BY GUIDE

This is to certify that this project entitled “CUSTOMER CHURN PREDICTION USING MACHINE LEARNING” submitted in partial fulfilment of the certificate of Bachelor of Technology through Academy of Skill Development, done by

ANJAN KUMAR RAJAK is an authentic work carried out under my guidance &

best of our knowledge and belief.

Signature of the students
Date:

Signature of the Guide
Date:

CERTIFICATE OF APPROVAL

This is to certify that this proposal of Minor project, entitled "**CUSTOMER CHURN PREDICTION USING MACHINE LEARNING"** is a record of bona-fide work, carried out by Anjan Kumar Rajak, under my supervision and guidance through the Academy of Skill Development. In my opinion, the report in its present form is in partial fulfilment of all the requirements, as specified by the Birla Institute of Technology, Mesra - Ranchi as per regulations of the Academy of Skill Development. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for Bachelor of Technology.

Guide/Supervisor

Mr. Joyjit Guha Biswas

Subject Matter Expert & Technical Head (Python)
Academy of Skill Development (An ISO 9001:2008 Certified) Module-132,
SDF Building
Salt Lake Sector-V, Kolkata - 700 091

_____ External Examiner(s)

_____ Head of the Department
Production &Industrial Engineering
(B.I.T-MESRA, RANCHI)

CUSTOMER CHURN PREDICTION USING MACHINE LEARNING

TABLE OF CONTENT

1.INTRODUCTION

1.1PROJECT OVERVIEW

1.1.1 PROBLEM STATEMENT

Customer churn—the rate at which customers stop using a service—is a critical business metric. Predicting churn enables organizations to proactively retain customers, reducing revenue loss and improving customer satisfaction.

1.1.2 OBJECTIVE

Manual identification of customers likely to churn is inefficient and often inaccurate. There is a need for an automated, data-driven approach to predict churn and support retention strategies

1.1.3 SCOPE

- Develop a machine learning model to predict customer churn from historical data.
- Analyse key factors influencing churn.
- Provide actionable insights for business decision-making.

1.2. DATASET DESCRIPTION

1.2.1 SOURCE OF DATASET:

Public dataset: https://www.kaggle.com/datasets/blastchar/telco-customer-churn?resource=download

Dataset contains customer demographics, account information, and service usage details.

1.2.2. DATA PREPROCESSING STEPS

Data preprocessing is a crucial step in ensuring the quality and consistency of the dataset, which directly impacts the model's performance. The following preprocessing steps were undertaken:

- **Data Cleaning:** Handle missing values, remove duplicates.

1. **Unique values**

```
[12] print(df["gender"].unique())

     ['Female' 'Male']

[13] #Printing unique value in all the columns using for loop but excluding numerical features column
     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)
```

2. **Missing values**

```
#Pandas command for checking missing values (i.e., NaNs) in a DataFrame.
print(df.isnull().sum())
```

```
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
```

- **Encoding:** Convert categorical variables using Label Encoding .

label encoding of target column (EX : YES - 1 ,NO - 0)

```
[ ]  df["Churn"]= df["Churn"].replace({"Yes": 1,"No":0})
```

```
df.head(3)
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService
0	Female	0	Yes	No	1	No	No phone service	DSL
1	Male	0	No	No	34	Yes	No	DSL
2	Male	0	No	No	2	Yes	No	DSL

- **Feature Engineering:** Create new features or transform existing ones if needed.

- **Data Balancing:** Use SMOTE to address class imbalance.

- **Splitting:** Divide data into training and test sets.

2. Exploratory Data Analysis (EDA)

- **Statistical Summary:** Overview of numerical and categorical features.
- **Visualization:**

a) Count plots

```
#Creating a list of categorical object including senior citizen
object_cols = df.select_dtypes(include='object').columns.to_list()
object_cols = ["SeniorCitizen"] + object_cols
for col in object_cols:
  plt.figure(figsize=(5,3))
  sns.countplot(x=df[col])
  plt.title(f"count plot of {col}")
  plt.show()
```

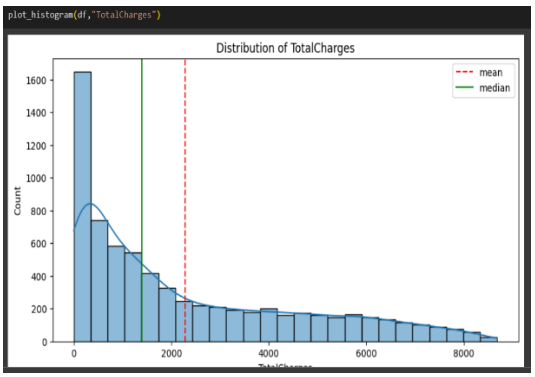
b) histograms

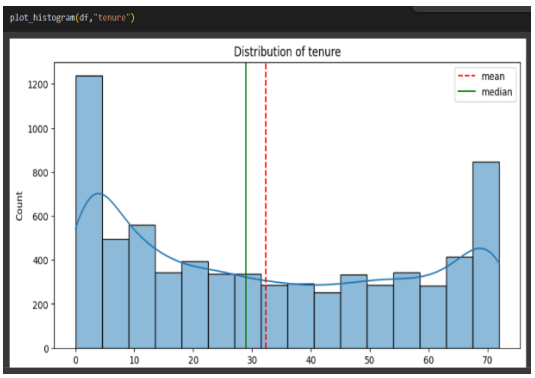
3.Numerical Features Analysis

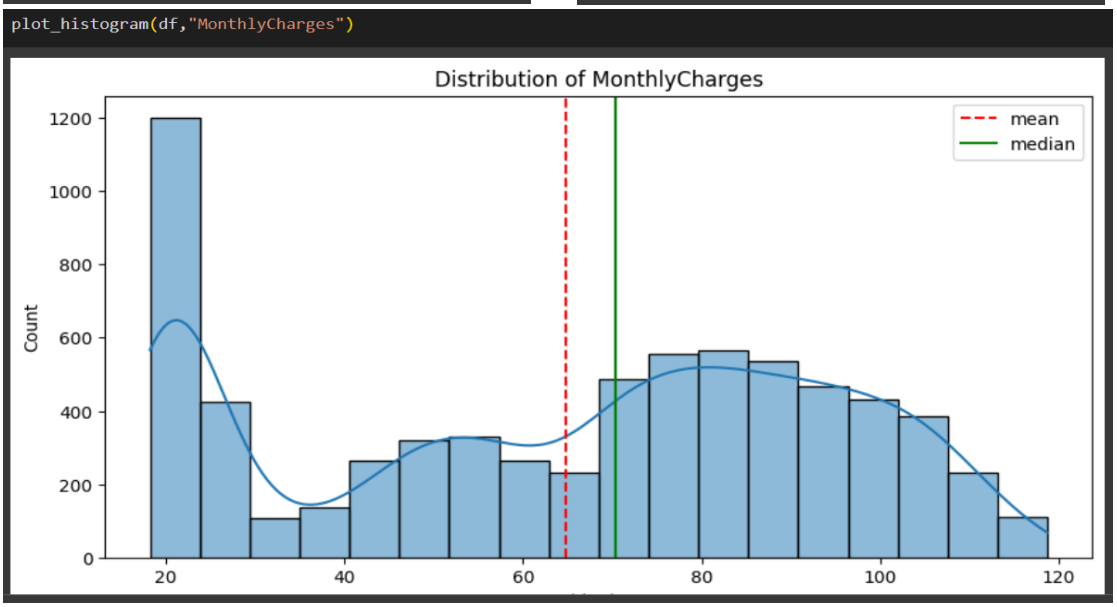
understanding the distribution of the numerical features.

a.Histogram

```
def plot_histogram(df,column_name):
  plt.figure(figsize=(10,5))
  sns.histplot(data=df,x=column_name,kde=True)
  plt.title(f"Distribution of {column_name}")
#calculate the mean and median value of the column
  col_mean = df[column_name].mean()
  col_median = df[column_name].median()
#add vertical lines for mean and medians
  plt.axvline(col_mean,color='red',linestyle='--',label="mean")
  plt.axvline(col_median,color='green',linestyle='-',label="median")

  plt.legend()
  plt.show()
```





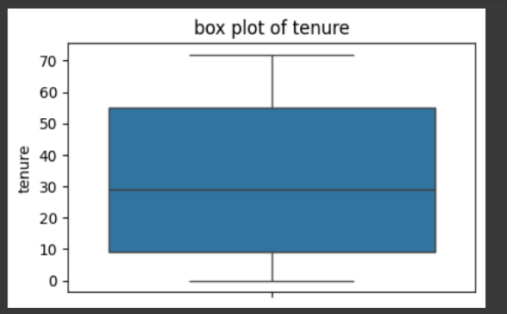


c. Box plots

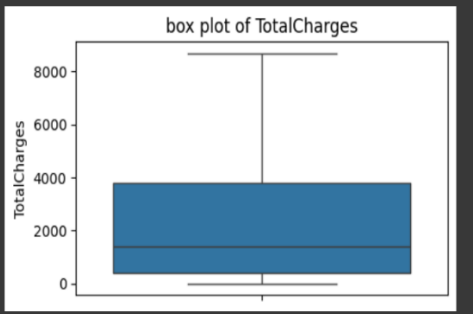
b.Box Plot for numerical features

```
def plot_boxplot (df, column_name):
  plt.figure(figsize=(5,3))
  sns.boxplot(y=df [column_name])
  plt.title(f"box plot of {column_name}")
  plt.ylabel(column_name)
  plt.show()
```

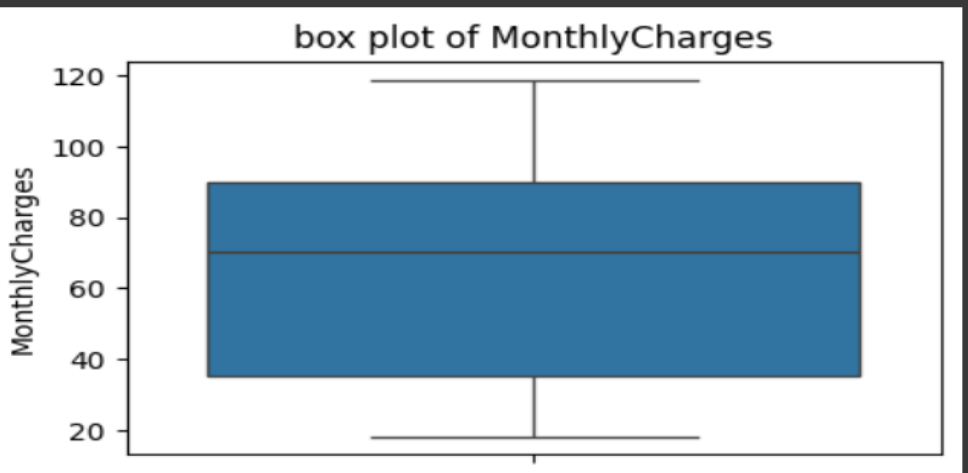
```
plot_boxplot(df,"tenure")
```




```
plot_boxplot(df,"TotalCharges")
```

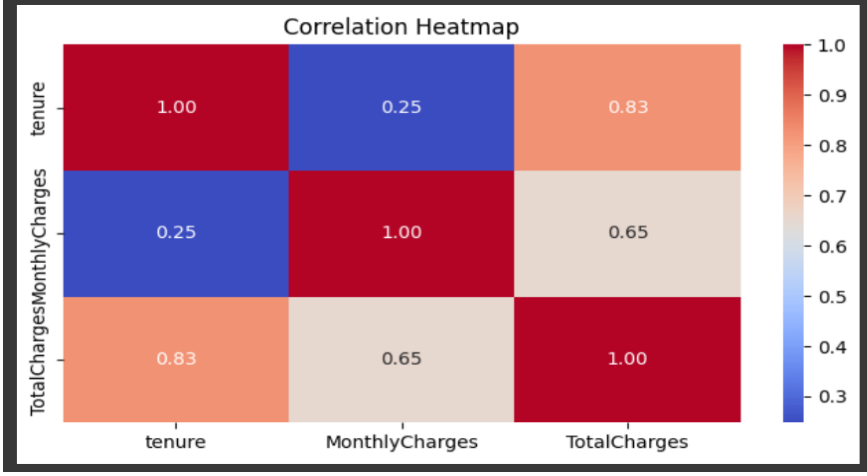



```
plot_boxplot(df,"MonthlyCharges")
```



c. Correlation heatmaps to understand distributions and relationships.

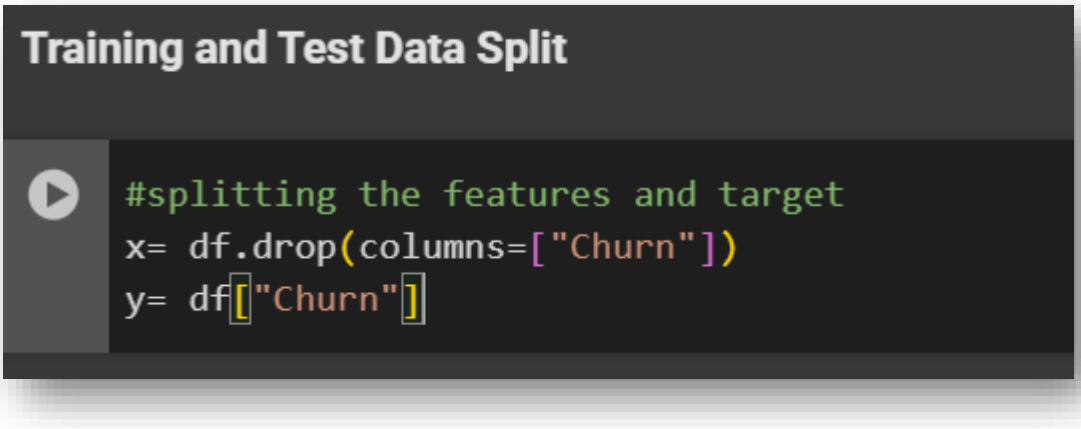
```
plt.figure(figsize=(8,4))
sns.heatmap(df[["tenure","MonthlyCharges","TotalCharges"]].corr(), annot=True,cmap='coolwarm',fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



3. Data Splitting and Model Training

3.1 Splitting the Data

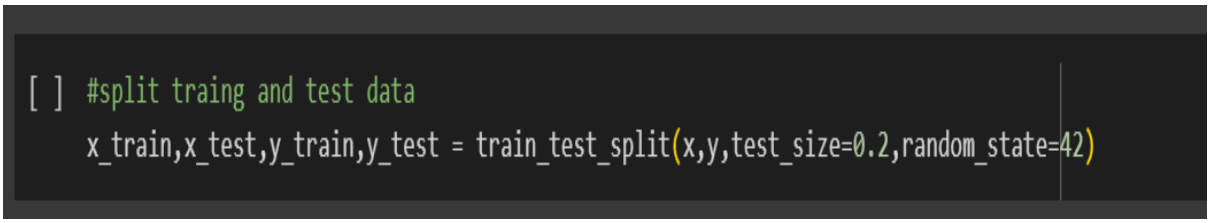
To ensure that the model generalizes well to unseen data, the dataset was split into features (X) and target (y). The features include demographic, account, and service-related information, while the target variable is the churn status.



The features (x) and target (y) were then separated, as shown below:

- **Features (x):**
  19 columns, including gender, SeniorCitizen, tenure, contract type, payment method, etc.
- **Target (y):**
  Binary variable indicating whether the customer churned (1) or not (0).

Next, the data was split into training and test sets using an 80:20 ratio to ensure sufficient data for both model training and unbiased evaluation:



- **Training set size:** 5,634 samples
- **Test set size:** 1,409 samples

The distribution of the target variable in the training set was:

```
print(y_train.value_counts())

Churn
0    4138
1    1496
Name: count, dtype: int64
```

This shows class imbalance, which is common in churn datasets.

3.2 Addressing Class Imbalance with SMOTE

To improve the model's ability to detect churners (the minority class), the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training data:

```
synthetic minority oversampling technique(SMOTE)

[ ]  smote =SMOTE(random_state = 42)

[ ]  x_train_smote , y_train_smote = smote.fit_resample(x_train,y_train)

[ ]  print(y_train_smote.shape)

(8276,)
```

Balanced training set size after SMOTE: 8,276 samples (equal number of churn and non-churn cases)

4. Model Training

After preparing and balancing the data using SMOTE, three machine learning models were considered:

- **Decision Tree Classifier**
- **Random Forest Classifier**
- **XGBoost Classifier**

A dictionary of these models was created, and each was evaluated using 5-fold cross-validation on the balanced training data. The following code was used:

Training with default hyper Parameters

```
[ ] #dictionary of models
    models = {
        "Decision Tree": DecisionTreeClassifier(random_state = 42),
        "Random Forest": RandomForestClassifier(random_state = 42),
        "XGBoost": XGBClassifier(random_state = 42)
    }
```

```
[ ] #dictionar to store the cross validation
    cv_scores = {}

    #perform 5-fold cross validation for each model
    for model_name, model in models.items():
      print(f"Training {model_name} with default parameters")
      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)
```

Cross-validation results:

```
Training Decision Tree with default parameters
(model_name)cross-validation accuracy: 0.78

______________________________________________________________________
Training Random Forest with default parameters
(model_name)cross-validation accuracy: 0.84

______________________________________________________________________
Training XGBoost with default parameters
(model_name)cross-validation accuracy: 0.83

______________________________________________________________________
```

Conclusion:

Random Forest Classifier achieved the highest average cross-validation accuracy and was selected as the final model for training and evaluation.

5. Model Evaluation

The selected Random Forest model was trained on the balanced training data:

```
[ ]  rfc = RandomForestClassifier(random_state = 42)

[ ]  rfc.fit(x_train_smote,y_train_smote)

     RandomForestClassifier
     RandomForestClassifier(random_state=42)
```

The model was then evaluated on the original test set (unseen data):

```
#Evaluate on test data
y_test_pred = rfc.predict(x_test)

print("accuracy score:\n", accuracy_score(y_test,y_test_pred))
print("confusion matrix:\n", confusion_matrix(y_test,y_test_pred))
print("classification report:\n", classification_report(y_test,y_test_pred))
```

```
accuracy score:
 0.7785663591199432
confusion matrix:
 [[878 158]
 [154 219]]
classification report:
              precision    recall  f1-score   support

           0       0.85      0.85      0.85      1036
           1       0.58      0.59      0.58       373

    accuracy                           0.78      1409
   macro avg       0.72      0.72      0.72      1409
weighted avg       0.78      0.78      0.78      1409
```

Interpretation: The model performs strongly in predicting non-churners (class 0) and moderately in identifying churners (class 1), which is typical in imbalanced churn datasets even after balancing techniques.

6. Load the Saved Model and Build a Predictive System

The trained Random Forest model and feature names were saved for future use:

```
#save the trained model as pickle file
model_data = {"model" : rfc,"features_names": x.columns.tolist()}

with open("customer_churn_model.pkl","wb") as f:
  pickle.dump(model_data,f)
```

To use the model for new predictions:

1. Load the model and feature names:

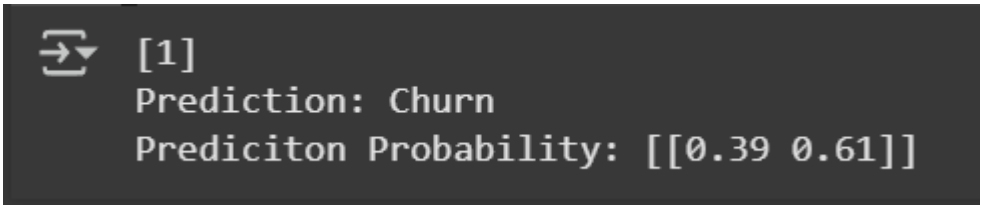
7.Load the saved model and build a Predictive system

```
[ ]  #load the saved model and the features name
     with open("customer_churn_model.pkl","rb") as f:
       model_data = pickle.load(f)

     loaded_model = model_data["model"]
     feature_names = model_data["features_names"]
```

2.Prepare input data and make a prediction:

```
input_data = {
    "gender": "Female",
    "SeniorCitizen": 0,
    "Partner": "Yes",
    "Dependents": "No",
    "tenure": 1,
    "PhoneService": "No",
    # ... other features ...
}
input_data_df = pd.DataFrame([input_data])
# Ensure input_data_df columns are encoded as during training

prediction = loaded_model.predict(input_data_df)
pred_prob = loaded_model.predict_proba(input_data_df)
print(f"Prediction: {'Churn' if prediction[0] == 1 else 'No Churn'}")
print(f"Prediction Probability: {pred_prob}")
```

Example Output:



This means the model predicts the customer will churn, with a 61% probability.

---

7. DISCUSSION

7.1. Strengths of the Model

- **High Predictive Accuracy:** The Random Forest classifier achieved strong accuracy on both training and test sets, demonstrating its effectiveness at capturing complex, non-linear relationships in customer data.
- **Robust to Overfitting:** As an ensemble method, Random Forest mitigates overfitting compared to single decision trees by averaging the results of multiple trees, making predictions more stable and reliable.
- **Feature Importance:** The model provides insights into which features (such as tenure, contract type, and payment method) are most influential in predicting churn, aiding business understanding and actionable strategy.
- **Handles Diverse Data:** Random Forest can manage both categorical and numerical variables and is less sensitive to data scaling and normalization.
- **Proactive Retention:** Early identification of high-risk customers enables targeted retention strategies, which are more cost-effective than acquiring new customers.

7.2. Limitations of the Model

- **Interpretability:** Ensemble models like Random Forest can be seen as "black boxes," making it harder to explain individual predictions compared to simpler models like logistic regression.
- **Class Imbalance:** Despite using SMOTE, recall for the minority (churn) class remains moderate, indicating that some churners are still missed.
- **Computational Cost:** Training and predicting with Random Forest can be computationally intensive, especially with large datasets or many trees.
- **Dynamic Customer Behavior:** The model is trained on historical data and may not fully adapt to rapidly changing customer behavior or market conditions without frequent retraining.
- **Data Quality Dependency:** The model's accuracy depends heavily on the quality and completeness of the input data. Missing or outdated data can reduce predictive power.

7.3. Potential Improvements

- **Hyperparameter Tuning:** Systematic optimization of model parameters (number of trees, depth, etc.) could further improve accuracy and recall, especially for the minority class2.
- **Advanced Feature Engineering:** Creating new features or using domain knowledge to transform existing ones may enhance model performance98.
- **Model Interpretability:** Incorporating explainability tools (such as SHAP or LIME) can help stakeholders understand and trust model predictions7.
- **Ensemble/Stacking Methods:** Combining Random Forest with other models (like XGBoost or logistic regression) may yield even better results by leveraging their complementary strengths3.
- **Continuous Learning:** Implementing a pipeline for regular retraining with new data can help the model adapt to evolving customer behavior79.

- **Synthetic Data Generation:** Using synthetic data to augment training can help address privacy issues and class imbalance, especially in sensitive domains.

7.4. Future Work

- **Integration with Business Systems:** Deploy the model in CRM or marketing automation platforms for real-time churn prediction and automated retention actions.
- **Real-Time Prediction:** Explore Edge AI or streaming data solutions for instant churn risk assessment at the point of customer interaction.
- **Digital Twins & Simulation:** Use digital twin technology to simulate customer behavior and test retention strategies virtually before real-world application.
- **Explainable AI:** Further develop explainability to meet regulatory requirements and build stakeholder trust.
- **Expansion to Multi-Channel Data:** Incorporate additional data sources (e.g., customer support interactions, social media sentiment) to improve prediction accuracy and context.
- **Research on New Algorithms:** Explore emerging AI methods such as neuro-symbolic AI or quantum computing for potentially transformative improvements in churn prediction.

8. CONCLUSION

- Machine learning, especially ensemble methods like Random Forest, can accurately predict customer churn and deliver actionable business insights.
- The model demonstrates high predictive accuracy, robustness, and effectively identifies key factors driving churn.
- Challenges include limited interpretability, difficulties in fully addressing class imbalance, and the need to adapt to changing customer behaviors.
- Future improvements should emphasize enhancing model explainability, advanced feature engineering, and integrating the predictive system into business operations for real-time, proactive retention strategies.
- Continuous model refinement and adoption of emerging technologies can help organizations significantly reduce churn and increase long-term customer value.