**Telco Customer Churn Prediction - Project Report**

***Ganateja Thurpati***

***Manideep Reddy Janga***

**Project Overview:**

The objective of this project was to develop a machine learning model to predict customer churn using the Telco Customer Churn dataset. The dataset contains 7,043 customer records and 21 features, including demographics, service details, and billing information.

Churn prediction is critical in the telecom industry, as retaining customers is much more cost-effective than acquiring new ones. This project aimed to build a predictive model and analyze the key factors that influence churn.

**Data Preprocessing**

1. Loaded the dataset (WA\_Fn-UseC\_-Telco-Customer-Churn.csv).
2. Checked for duplicate records and verified that none existed.
3. Converted TotalCharges from object to numeric; handled 11 missing values by dropping those rows.
4. Created a tenure\_group column by binning tenure into 12-month intervals to simplify tenure interpretation.
5. Dropped non-predictive columns like customerID as it holds no useful information for modeling.
6. Analyzed categorical and numerical columns separately to understand value distributions.
7. One-hot encoded all categorical features to prepare the data for model input.
8. Standardized numerical columns (MonthlyCharges, TotalCharges) for future PCA-based modeling.
9. Split the dataset into features (X) and target (y = Churn).
10. Verified there were no remaining missing values post-transformation using isnull().sum().
11. Loaded the dataset (WA\_Fn-UseC\_-Telco-Customer-Churn.csv).
12. Converted TotalCharges to numeric and handled 11 missing values using row removal.
13. Created a tenure\_group column by binning tenure into 12-month intervals.
14. Dropped non-predictive columns like customerID.
15. One-hot encoded categorical features.
16. Split the data into features (X) and target (y = Churn).

**Handling Class Imbalance**

To address class imbalance, we used SMOTEENN, which combines Synthetic Minority Oversampling Technique (SMOTE) with Edited Nearest Neighbors (ENN). This helps both oversample the minority class and clean noisy examples.

Before applying SMOTEENN, the target variable distribution was significantly imbalanced:

Churn = 0 (No Churn): 5174

Churn = 1 (Churn): 1869

This imbalance can bias the model to favor the majority class (no churn), reducing the model’s ability to detect churners.

We chose SMOTEENN over simpler resampling strategies like random oversampling or SMOTE alone because:

* SMOTE creates synthetic minority class examples that help balance the dataset.
* ENN removes ambiguous or overlapping samples from both classes, improving class separation.

from imblearn.combine import SMOTEENN

sm = SMOTEENN()

X\_resampled1, y\_resampled1 = sm.fit\_resample(X, y)

After resampling:

Churn = 1: 3138

Churn = 0: 2653

This resampling approach helped the model learn from a more balanced dataset while reducing noise, which led to stronger generalization performance on unseen data. To address class imbalance, we used SMOTEENN, which combines Synthetic Minority Oversampling Technique (SMOTE) with Edited Nearest Neighbors (ENN). This helps both oversample the minority class and clean noisy examples.

from imblearn.combine import SMOTEENN

sm = SMOTEENN()

X\_resampled1, y\_resampled1 = sm.fit\_resample(X, y)

After resampling:

Churn = 1: 3138

Churn = 0: 2653

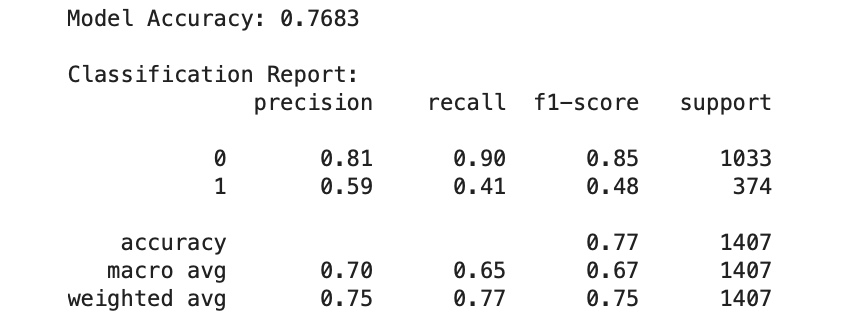
**Models Built**

**1. Decision Tree (Baseline Model)**

model\_dt = DecisionTreeClassifier(max\_depth=6, min\_samples\_leaf=8, random\_state=100)

model\_dt.fit(x\_train, y\_train)

**AUC Score:** 0.8101

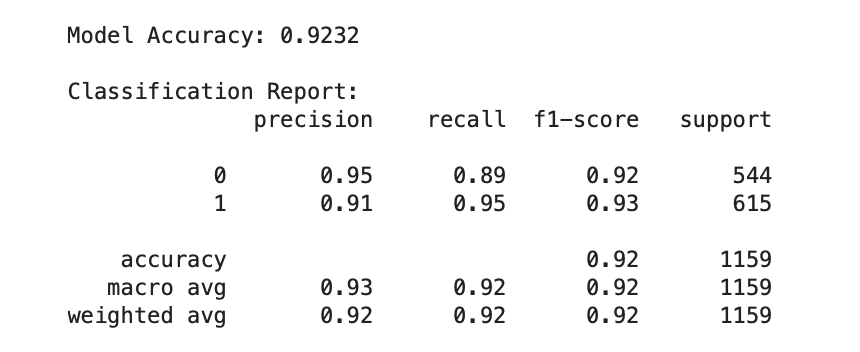


**2. Random Forest with SMOTEENN (Final Model)**

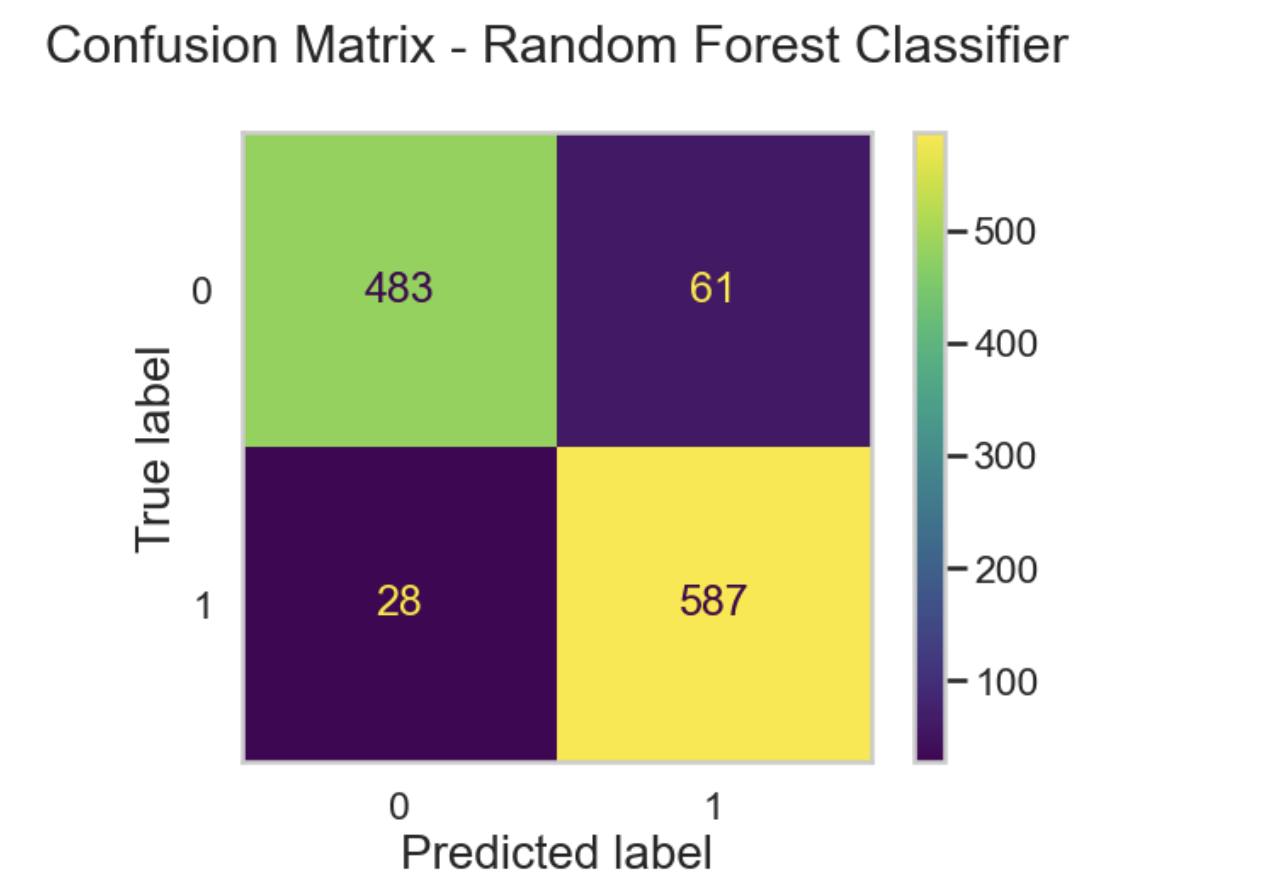
model\_rf\_smote = RandomForestClassifier(max\_depth=6, min\_samples\_leaf=8, n\_estimators=100, random\_state=100)

model\_rf\_smote.fit(xr\_train1, yr\_train1)

**Accuracy:** 0.9232 **AUC Score:** 0.9747

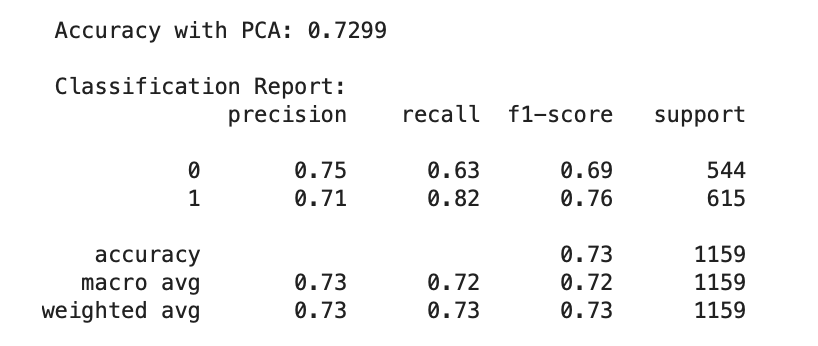


Confusion Matrix - Random Forest with SMOTEENN



**3. Random Forest with PCA**

Applied PCA to reduce dimensions while retaining 90% variance. **AUC Score:** 0.8213 (lower than the full-feature RF model)



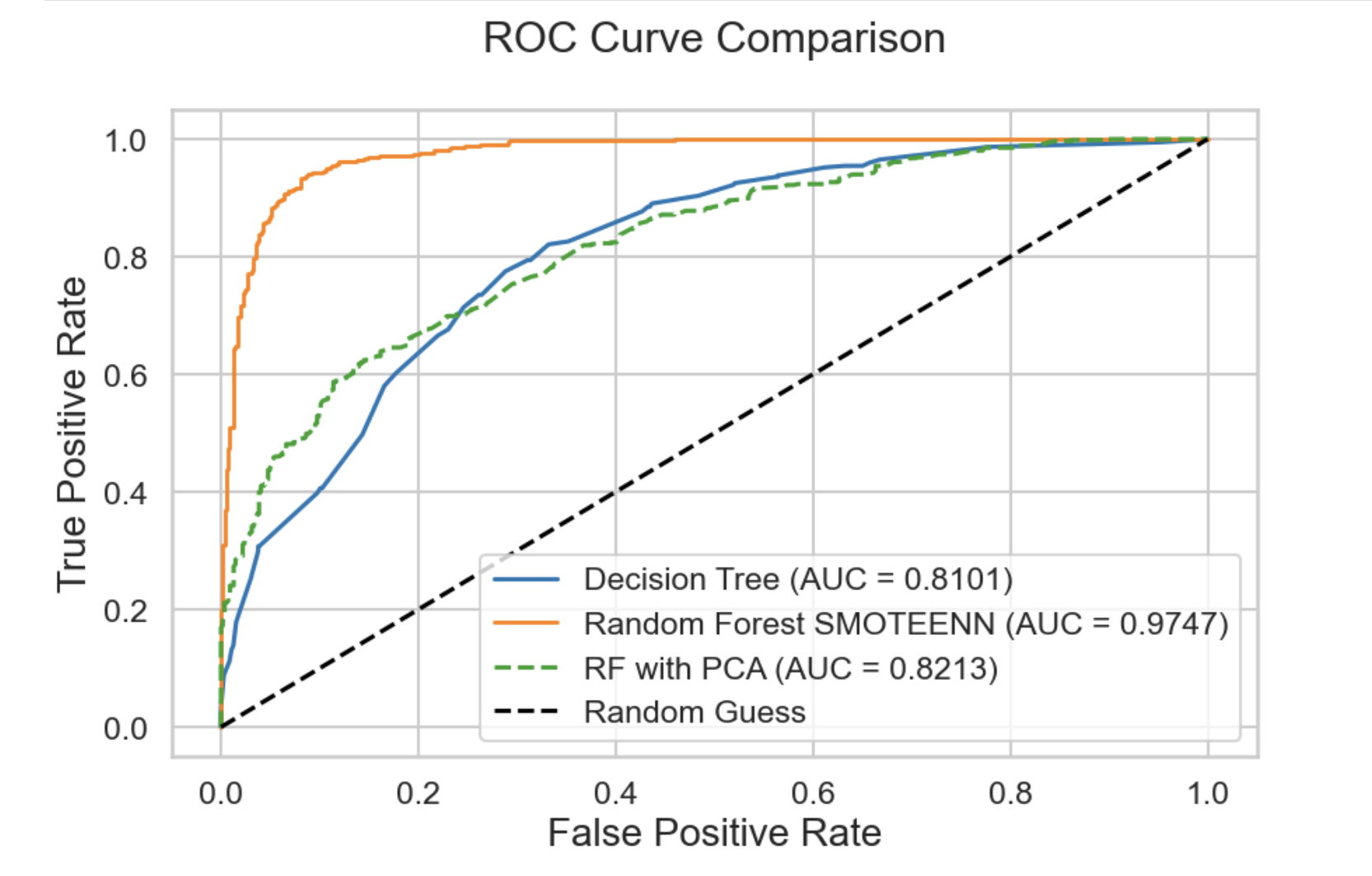
**ROC Curve Comparison**

The ROC Curve above compares the performance of three models:

|  |  |
| --- | --- |
| **Model** | **AUC Score** |
| Random Forest (SMOTEENN) | **0.9747** |
| Decision Tree | 0.8101 |
| Random Forest with PCA | 0.8213 |

* The **Random Forest model with SMOTEENN** achieves the **highest AUC (0.9747)**, clearly outperforming other models. This indicates excellent class separability and confirms it as the best-performing model.
* The **Decision Tree model**, though simpler, achieves a reasonable AUC of **0.8101**, but lacks the complexity to capture deeper patterns in the data.
* The **Random Forest model using PCA** shows slightly better performance than Decision Tree (**AUC = 0.8213**), but still lags behind the SMOTEENN-enhanced version.

The ROC comparison clearly shows that **Random Forest with SMOTEENN** is the most reliable model for customer churn prediction in this analysis.



**Feature Importance**

Understanding which features contribute most to model predictions is key for decision-making and interpretability. Random Forest allows us to rank features based on their importance in reducing impurity during training.

Top 10 features from Random Forest Classifier:

1. TotalCharges (0.165)

2. InternetService\_Fiber optic (0.120)

3. PaymentMethod\_Electronic check (0.115)

4. tenure\_group\_61 - 72 (0.096)

5. Contract\_Two year (0.087)

6. MonthlyCharges (0.053)

7. StreamingMovies\_No internet service (0.047)

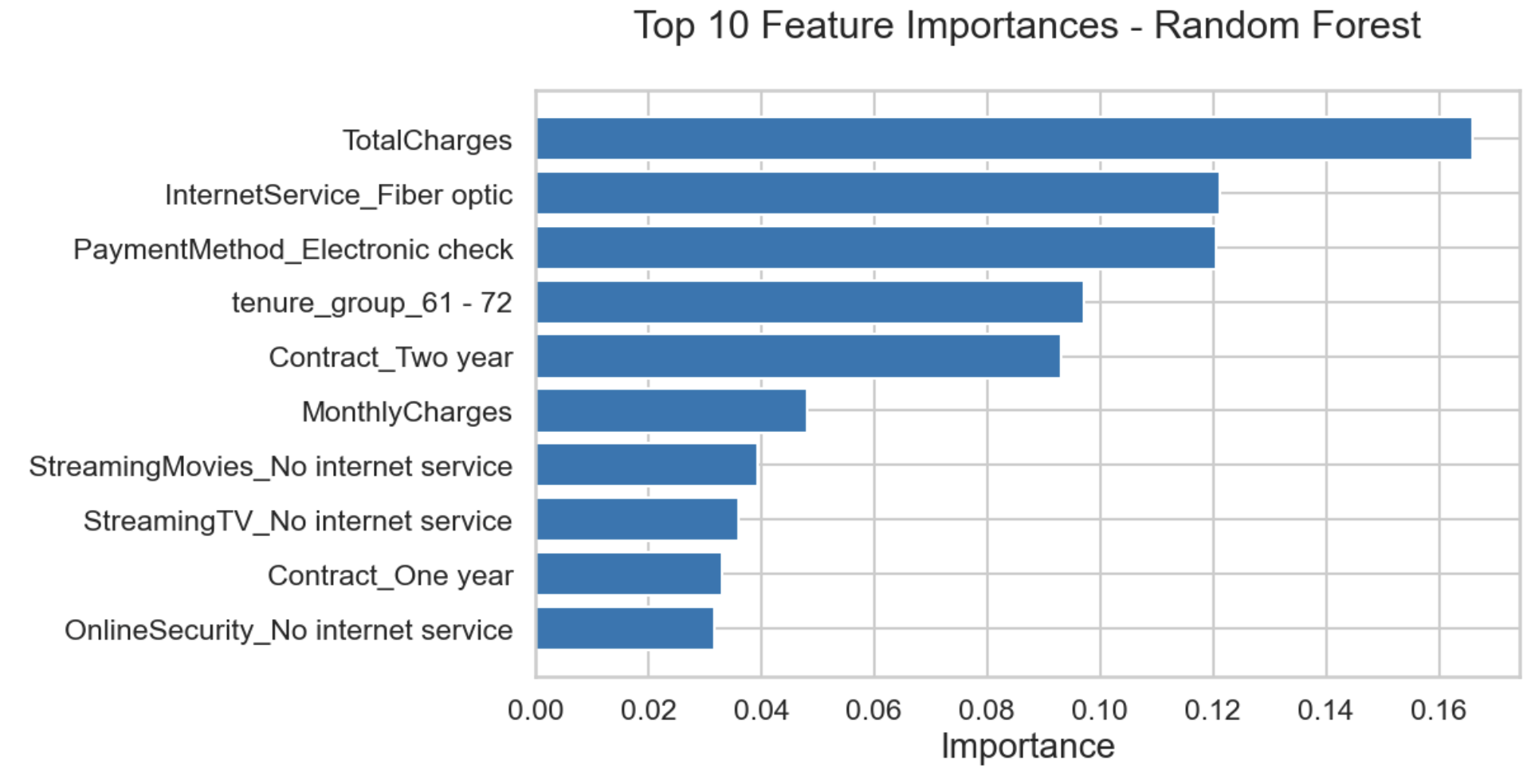
8. StreamingTV\_No internet service (0.045)

9. Contract\_One year (0.038)

10. OnlineSecurity\_No internet service (0.036)

These results show that:

* **TotalCharges** is the most informative predictor — lower charges are often linked to churn.
* Customers with **Fiber optic Internet** and **Electronic check** payments are highly prone to churn.
* Long tenure (**61–72 months**) and **2-year contracts** are strong retention indicators.



**Key Insights & Saving the Final Model**

**Key Insights**

* **Higher churn** is associated with:
  + Use of **fiber optic internet**
  + **Electronic check** as a payment method
  + **Short tenure** or customers in early contract stages
* **Lower churn** is seen in:
  + Customers with **long-term contracts** (1–2 years)
  + Customers with **lower total charges**

**Saving the Final Model**

import pickle

pickle.dump(model\_rf\_smote, open('model.sav', 'wb'))

To load the model:

model = pickle.load(open('model.sav', 'rb'))

**References**

* Dataset: [Kaggle - Telco Customer Churn](https://www.kaggle.com/blastchar/telco-customer-churn)
* SMOTEENN: [imblearn Documentation](https://imbalanced-learn.org/stable/)
* ROC Curve: scikit-learn ROC utilities