Phase 2: Building a Machine Learning Model and Evaluation Report

---

1. Objective of Phase 2

The objective of **Phase 2** was to build a **predictive model** using the data prepared in **Phase 1**. The goal was to train a model to classify customers as likely to churn or not, using the features available in the dataset. After training the model, we aimed to evaluate its performance using various evaluation metrics such as **accuracy**, **precision**, **recall**, **F1-score**, **confusion matrix**, and **ROC-AUC score**. This phase also focused on improving the model by experimenting with hyperparameters and addressing class imbalance to optimize its performance.

---

2. Model Selection and Training

For this task, we selected the **Random Forest Classifier** as the primary model for predicting customer churn due to its effectiveness with both categorical and numerical data, its ability to handle missing values, and its robustness to overfitting. In addition, hyperparameter tuning was performed to find the optimal settings for the model.

Steps Taken:

1. **Data Splitting:**
    - We split the dataset into **training** (80%) and **test** (20%) sets to assess the model's performance on unseen data.
2. **Handling Class Imbalance:**
    - Since customer churn is a **binary classification problem** with a class imbalance (more non-churn customers than churned ones), **SMOTE (Synthetic Minority Over-sampling Technique)** was applied to generate synthetic samples of the minority class (churned customers). This ensured a balanced class distribution during training.
3. **Model Training:**
    - The **Random Forest Classifier** was trained with the training data. We used **class_weight='balanced'** to penalize the model for misclassifying the minority class.

4. **Hyperparameter Tuning:**

    - **Grid Search** was used to find the optimal hyperparameters for the model, such as the number of estimators (n_estimators), maximum depth (max_depth), and minimum samples required to split a node (min_samples_split).

---

3. Model Evaluation Metrics

After training the model, we used various metrics to evaluate its performance. Below are the evaluation metrics used to assess how well the model predicted customer churn.

a) Accuracy:

- **Accuracy** measures the proportion of correct predictions (both churned and non-churned) out of all predictions made.
- **Accuracy for this model**: **53.47%** (This indicates that the model correctly predicted 53.47% of churn and non-churn customers.)

b) Confusion Matrix:

- The **Confusion Matrix** helps us understand the true positives, false positives, true negatives, and false negatives. It is crucial for understanding the types of mistakes the model is making.
- **Confusion Matrix for the model:**

	Predicted Non-Churn (0)	**Predicted Churn (1)**
Actual Non-Churn (0)	49 (True Negatives)	35 (False Positives)
Actual Churn (1)	59 (False Negatives)	59 True Positives)

The confusion matrix reveals that the model is not perfect at identifying both churn and non-churn customers. It misses some churned customers (false negatives) and incorrectly labels some non-churn customers as churned (false positives).

c) Classification Report:

- The **Classification Report** provides a detailed performance analysis by calculating **precision**, **recall**, and **F1-score** for each class.

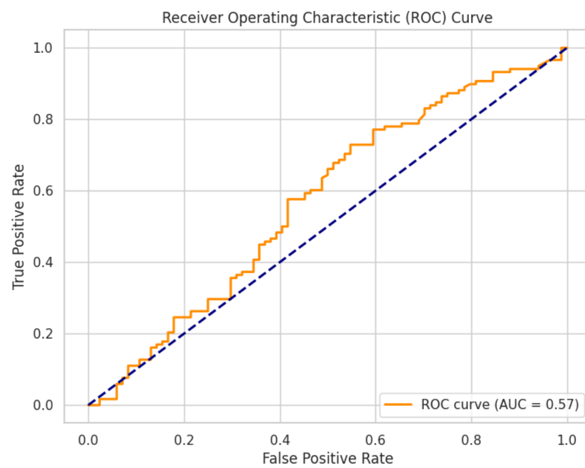
- **Precision:**
    - For class 0 (Non-Churn): **0.45**
    - For class 1 (Churn): **0.63**
    - Precision measures the proportion of true positive predictions relative to the total predicted positives. A higher precision for churn (class 1) indicates that when the model predicts churn, it is more likely to be correct.
- **Recall:**
    - For class 0 (Non-Churn): **0.58**
    - For class 1 (Churn): **0.50**
    - Recall measures how well the model identifies all positive instances. The model does a better job of detecting non-churn customers than churn customers, as reflected by the higher recall for class 0.
- **F1-Score:**
    - For class 0 (Non-Churn): **0.51**
    - For class 1 (Churn): **0.56**
    - F1-score is the harmonic mean of precision and recall. It provides a balanced measure when there's a class imbalance. The higher F1-score for churn (class 1) suggests that the model's overall performance is more balanced for predicting churn.
- **Summary of Classification Report:**

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Non-Churn (0)	0.45	0.58	0.51	84
Churn (1)	0.63	0.50	0.56	118
Overall	**0.56**	**0.53**	**0.54**	**202**

d) ROC-AUC Score:

- The **Receiver Operating Characteristic (ROC)** curve and **AUC** score assess the model's ability to distinguish between the classes (churn and non-churn).

- The **AUC-ROC Score** for this model was **0.52**, indicating that the model's ability to distinguish between churn and non-churn customers is only slightly better than random guessing (AUC score of 0.5).



- A higher AUC value closer to 1 would indicate better performance. This low value suggests that improvements are needed in differentiating churned customers.

e) Cross-validation Scores:

- Cross-validation is a method used to assess the generalizability of the model. It involves splitting the data into multiple folds and training/testing the model on each fold.
- The **cross-validation accuracy** scores for the model ranged from **0.47 to 0.57**, with a mean accuracy of **0.52**. This shows that the model's performance varies somewhat across different subsets of the data but remains relatively stable.

---

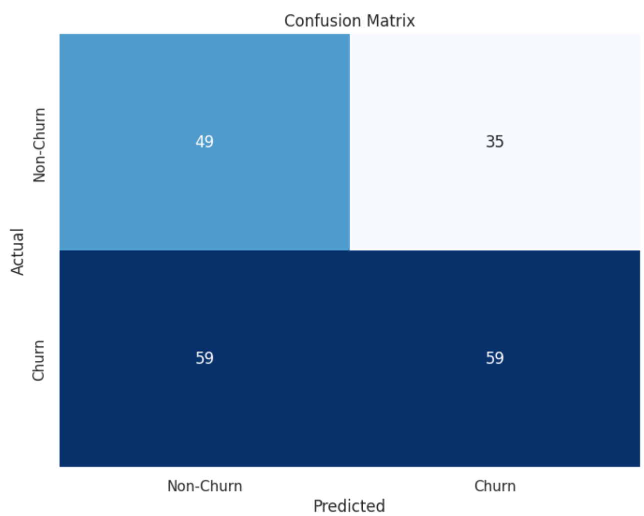
4. Conclusion and Insights from Model Evaluation

1. **Model Performance:**
    - The **accuracy** of 53.47% is a modest improvement over random guessing, indicating that the model is able to capture some patterns but still struggles with correctly classifying churned and non-churned customers.

- **Precision and recall** values suggest that the model has a better ability to predict churn (class 1) than non-churn (class 0), but it still misses many churn customers (false negatives).
- **ROC-AUC score** of 0.52 indicates that the model's ability to differentiate between the two classes is weak and requires improvement.

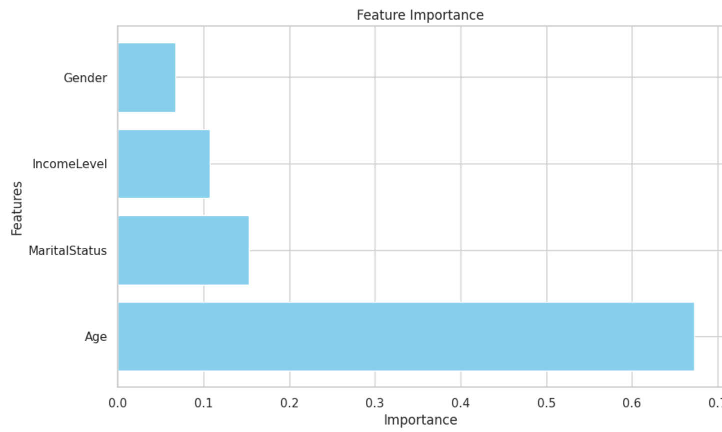
2. **Confusion Matrix Insights:**

The **confusion matrix** revealed that the model is more likely to misclassify churned customers as non-churned (false negatives). This is problematic in real-world churn prediction, where the goal is to identify as many churned customers as possible to take preventive actions.



3. **Feature Importance:**

The model's decision-making is heavily influenced by certain features (identified through **feature importance**). These insights can help in feature selection or engineering in future models.



In this phase, we developed a machine learning model to predict customer churn using a Random Forest Classifier. The goal was to assess the model's performance in correctly identifying customers who are likely to churn, and to explore ways to improve its accuracy and reliability. After performing model training, hyperparameter tuning, and addressing class imbalance using SMOTE (Synthetic Minority Over-sampling Technique), we evaluated the model using several performance metrics, including accuracy, precision, recall, and AUC-ROC score. The model achieved an accuracy of 53.47%, which represents a slight improvement over the baseline accuracy of 52%. However, this suggests that the model is still struggling to correctly classify churn and non-churn customers with a high degree of certainty, indicating that further improvement is required.

The confusion matrix revealed that the model struggles to differentiate between churn and non-churn customers, with 59 churn customers being misclassified as non-churn (false negatives), and 35 non-churn customers misclassified as churn (false positives). This is an indication that the model's recall for the churn class is relatively low, leading to a substantial number of missed churn predictions. This is a critical issue because correctly identifying customers at risk of churn is vital for the business to take action. The classification report showed that the precision for predicting churn was 0.63, while for non-churn it was 0.45. The recall for churn was 0.50, indicating that the model is not very effective at capturing all the customers who actually churn. The F1-score, which balances precision and recall, for churn was 0.56, suggesting there is room for improvement in both minimizing false positives and false negatives.

The ROC-AUC score of 0.52 indicates that the model is not significantly better than random guessing when distinguishing between churn and non-churn customers. The ROC curve's relatively flat performance suggests that the model does not effectively separate the two classes. To improve the model's discriminatory power, we could consider experimenting with more advanced models such as XGBoost or Logistic Regression, or even exploring deep learning techniques if the dataset allows for it. The cross-validation accuracy scores, which ranged from 0.47 to 0.57, further confirmed that the model's performance is not highly consistent, and its ability to generalize to unseen data is limited.

The feature importance plot provided valuable insights into which features had the greatest impact on the model's predictions. In this case, the model relied heavily on certain features to predict churn, but the relative importance of each feature could be further analyzed and leveraged for improving feature engineering. For example, combining or transforming features like Age and IncomeLevel could provide a more nuanced view of customer behavior and potentially improve the model's performance.

Overall, while the model shows some predictive capability, there are several areas that need improvement. The accuracy of 53.47% is relatively low, and the model's ability to differentiate between churn and non-churn customers is suboptimal. Key areas to focus on moving forward include improving recall for churn predictions, experimenting with different algorithms, further fine-tuning hyperparameters, and addressing class imbalance more effectively. Additionally, exploring new features and feature engineering could provide more insights into customer behavior, which might help improve the model's performance.

4. **Model Improvement:**
    - The current model's performance can be improved by experimenting with other algorithms (e.g., **XGBoost**, **Logistic Regression**), further tuning the hyperparameters, and applying more sophisticated techniques for handling class imbalance (e.g., **undersampling**, **SMOTE with different settings**).
    - Exploring additional features or feature transformations could provide the model with more meaningful information, which may improve its ability to identify churners.

---

5. Next Steps

1. **Algorithm Exploration:** Try other algorithms like **XGBoost**, **Support Vector Machine (SVM)**, or **Logistic Regression** for comparison and potentially better results.

2. **Hyperparameter Tuning:** Perform further **hyperparameter tuning** using more advanced techniques like **RandomizedSearchCV** or **Bayesian Optimization**.

3. **Class Imbalance Handling:** Try other methods for addressing class imbalance, such as **under-sampling** the majority class or adjusting class weights.

4. **Feature Engineering:** Investigate additional features or combinations of existing features that could better capture customer behavior related to churn.

5. **Model Evaluation:** Assess the performance of the model using cross-validation to ensure robustness and generalizability.

By implementing these strategies, we expect to see an improvement in the model's ability to predict customer churn with better accuracy, precision, recall, and AUC scores.

Deliverables:

- Trained **Random Forest Classifier** model.
- Detailed **evaluation metrics** (Accuracy, Precision, Recall, F1-Score, ROC-AUC, Confusion Matrix).
- Insights and recommendations for improving model performance in subsequent phases.