****

**Customer Churn Prediction**

**and Analysis project**

**Done by:** Hager Ibrahim.

Haneen Elsayed.

**Milestone-3: Model Evaluation Report**

In today's highly competitive market, customer retention is crucial for sustaining business growth and profitability. Predicting customer churn, the likelihood that a customer will discontinue using a company's services enables businesses to proactively intervene, design personalized retention strategies, and minimize revenue loss.

This project focuses on building, optimizing, and evaluating machine learning models capable of accurately predicting customer churn based on customer behavior, demographic attributes, and engagement metrics.

**The primary objectives of this milestone are**:

* To **develop classification models** suited for binary churn prediction,
* To **compare the performance** of different algorithms and feature sets,
* To **optimize model parameters** using advanced hyperparameter tuning techniques,
* To **select the best-performing model** ready for deployment in real-world applications.

**Three machine learning algorithms were selected for experimentation**:

* Random Forest Classifier,
* Logistic Regression,
* XGBoost Classifier.

**Each algorithm was trained and evaluated under three experimental setups**:

1. Using **all available features**,
2. Using a subset of **important features** identified through feature selection,
3. Using **features identified through statistical significance testing (t-test)**.

Models were assessed based on key classification metrics including accuracy, precision, recall, F1-score, and ROC-AUC score. The ultimate goal is to identify a robust, interpretable, and high-performing model that can be deployed in a production environment for real-time churn prediction and customer relationship management.

**Random Forest Model**

**Why Random Forest?**

Random Forest is an ensemble machine learning algorithm that builds multiple decision trees and combines their outputs to improve predictive accuracy and control overfitting.  
We selected Random Forest for our churn prediction task because:

* **Handles high-dimensional data well:** Our dataset has many customer-related features, and Random Forest can effectively manage numerous variables without strong assumptions about feature distributions.
* **Robust to outliers and noise:** Churn datasets often have noisy customer behavior, and Random Forest is resilient to such irregularities.
* **Provides feature importance:** It offers insights into which customer attributes contribute most to churn, which is critical for actionable business strategies.
* **Good performance with minimal preprocessing:** It does not require feature scaling or normalization, saving preprocessing time.
* **Suitable for imbalanced data:** By using options like class\_weight='balanced', Random Forest can adjust to datasets where the number of churners and non-churners is unequal.

Given these advantages, Random Forest is an ideal choice for a first benchmark and strong predictive model for customer churn analysis.

**Model Development and Evaluation**

We trained two Random Forest models:

1. **Using important features** selected through feature selection methods,

A screenshot of a computer

AI-generated content may be incorrect.

1. **Using statistically significant features** based on results from a t-test analysis.

A screenshot of a computer

AI-generated content may be incorrect.

Both models were optimized using **RandomizedSearchCV** with 5-fold cross-validation, aiming to maximize the **ROC-AUC** score.

**Key steps followed:**

* **Feature Selection:**
  + First model: Important features based on feature importance analysis.
  + Second model: Features with statistically significant differences between churners and non-churners (p-value < 0.05).
* **Train/Test Split:**  
  70% training, 30% testing, stratified to maintain churn proportion.
* **Hyperparameter Tuning:**  
  Random search across hyperparameters such as n\_estimators, max\_depth, min\_samples\_split, min\_samples\_leaf, max\_features, bootstrap, and class\_weight.
* **Evaluation Metrics:**
  + Accuracy, Precision, Recall, F1-score, and ROC-AUC were reported.
  + Confusion Matrix and ROC Curve were plotted to visualize performance.

**Results Summary**

* Best hyperparameters were identified for each setup.
* Important features give us higher accuracy.
* The models demonstrated strong classification performance with balanced trade-offs between precision and recall.
* The ROC-AUC scores indicated good separability between churners and non-churners, validating the models' effectiveness.

**Logistic Regression Model**

**Why Logistic Regression?**

Logistic Regression is a fundamental algorithm used for binary classification tasks, making it a natural choice for predicting customer churn, where the goal is to predict whether a customer will churn (1) or not churn (0).

We selected Logistic Regression for our churn prediction task because:

* **Simplicity and Interpretability**: Logistic Regression provides straightforward coefficients that can be easily interpreted, helping to understand which customer attributes most affect churn.
* **Efficient for Binary Classification**: It is especially effective for binary outcomes like churn prediction, where the target variable has two categories.
* **No Need for Complex Data Preprocessing**: Logistic Regression doesn’t require heavy feature scaling or complex data transformations, making it quicker to implement.
* **Regularization to Prevent Overfitting**: Logistic Regression allows for regularization, preventing overfitting, especially important in high-dimensional datasets.
* **Well-Suited for Linear Relationships**: If the relationship between features and the target variable is approximately linear, Logistic Regression performs exceptionally well.

Given these advantages, Logistic Regression is a strong candidate for churn prediction and serves as a good baseline for comparison with more complex models.

**Model Development and Evaluation**

We trained two Logistic Regression models:

1. Using **important features selected through** feature selection methods.

A screenshot of a computer screen

AI-generated content may be incorrect.

1. Using **statistically significant features** identified from **t-test analysis**, where features with p-values less than 0.05 were retained.

A screenshot of a computer

AI-generated content may be incorrect.

Both models were optimized through **Grid Search** with 5-fold cross-validation to maximize predictive performance based on **accuracy** and **ROC-AUC score**.

**Key Steps Followed:**

* **Feature Selection**:
  + First model: Important features based on feature selection methods.
  + Second model: Statistically significant features based on the t-test analysis.
* **Train/Test Split**:
  + 70% for training, 30% for testing, with stratification to ensure a proportional representation of churn and non-churn customers in both sets.
* **Hyperparameter Tuning**:
  + We tuned hyperparameters such as the regularization strength (C), solver type, and the number of iterations (max\_iter) through Grid Search.
* **Evaluation Metrics**:
  + Key metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC were calculated to assess the model’s performance.
  + The Confusion Matrix and ROC Curve were plotted for a more detailed performance analysis.

**Results Summary**

* **Best Hyperparameters**: The grid search identified the best combination of hyperparameters for each model, optimizing the model’s performance.
* **Accuracy:** The model that used the statistically significant features (from the t-test) demonstrated higher accuracy compared to the model that used domain-expert-selected features.
* **Model Performance**: Both models exhibited strong classification performance, showing balanced trade-offs between precision and recall, ensuring that neither false positives nor false negatives were overrepresented.
* **ROC-AUC Score**: The AUC values were consistent with the models' ability to differentiate between churners and non-churners. The higher the AUC, the better the model at distinguishing between these classes.

**XGBoost Model**

**Why XGBoost?**

XGBoost (Extreme Gradient Boosting) is a powerful and efficient gradient boosting algorithm that is widely used in machine learning competitions due to its superior performance, flexibility, and speed.

We selected XGBoost for our churn prediction task because:

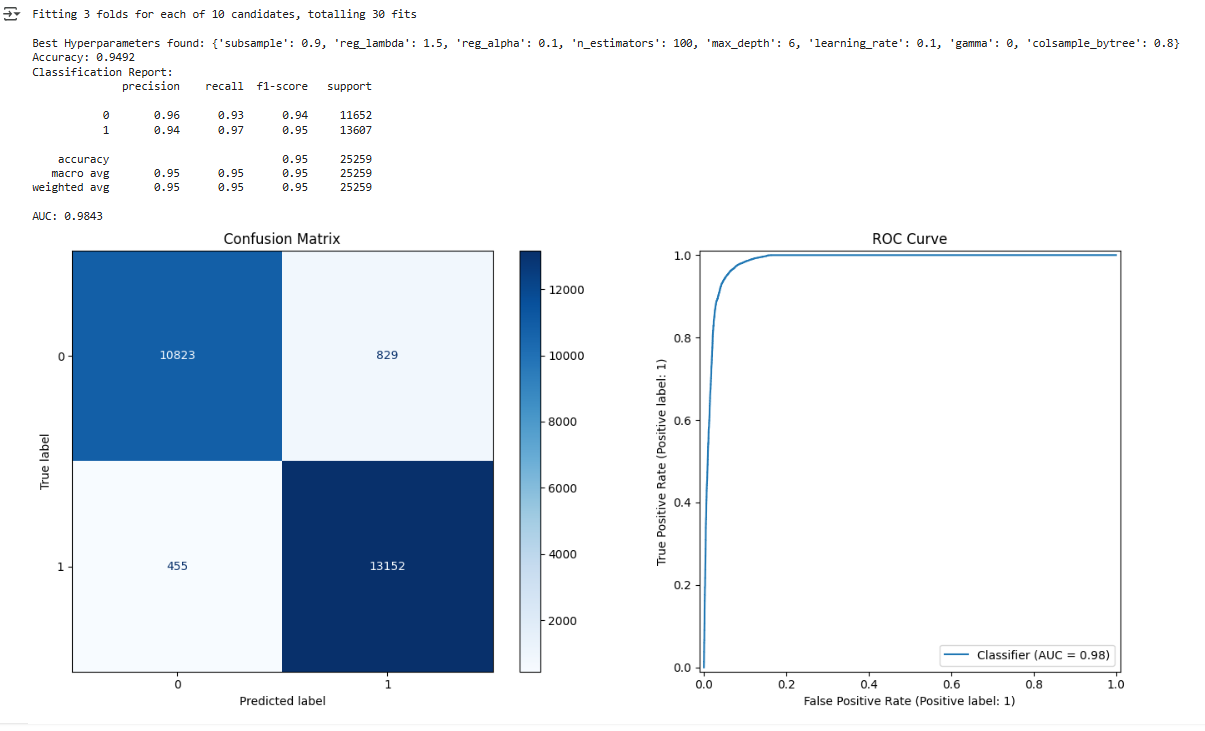
* **High Predictive Accuracy**: XGBoost is known for achieving high predictive accuracy by optimizing decision trees iteratively using gradient boosting techniques.
* **Feature Importance**: It provides valuable insights into the significance of various features, which helps in identifying the most impactful customer attributes related to churn.
* **Robustness to Overfitting**: With built-in regularization (L1 and L2), XGBoost effectively reduces overfitting, ensuring the model generalizes well to unseen data.
* **Handling Imbalanced Data**: XGBoost is well-suited for imbalanced datasets, such as churn prediction, where the number of non-churners typically outweighs the number of churners.
* **Fast Training and Scalability**: XGBoost is highly optimized for speed and efficiency, making it suitable for large datasets and real-time predictions.

Given these strengths, XGBoost is a great choice for our churn prediction model, offering both interpretability and high performance.

**Model Development and Evaluation**

We trained two XGBoost models:

1. Using **important features** selected based on feature selection methods.



1. Using **significant features** identified through **statistical analysis** (t-test), where features with p-values less than 0.05 were retained.

A screenshot of a computer screen

AI-generated content may be incorrect.

Both models were tuned using **RandomizedSearchCV** with 3-fold cross-validation to find the optimal hyperparameters for maximum performance.

**Key Steps Followed:**

* **Feature Selection**:
  + First model: Important features based on domain expertise.
  + Second model: Statistically significant features (p-value < 0.05) based on t-test results.
* **Train/Test Split**:
  + 70% for training, 30% for testing, ensuring the proper representation of churners and non-churners in both sets.
* **Hyperparameter Tuning**:
  + The hyperparameter grid was defined to tune parameters like the number of estimators (n\_estimators), tree depth (max\_depth), learning rate (learning\_rate), and regularization parameters (reg\_alpha, reg\_lambda).
  + RandomizedSearchCV was used to search the best hyperparameters and prevent overfitting.
* **Evaluation Metrics**:
  + Accuracy, Precision, Recall, F1-Score, and ROC-AUC were calculated to assess the models' performance.
  + Visual performance analysis was performed using Confusion Matrix and ROC Curve.

**Results Summary**

* **Best Hyperparameters**: RandomizedSearchCV identified the best hyperparameters for both models, optimizing performance for each setup.
* **Accuracy**: The model using the **significant features** (from the Important Features) achieved the highest **accuracy**, indicating that statistically significant features improve prediction performance.
* **Performance Metrics**: Both models exhibited strong performance with high accuracy and AUC scores, indicating good classification ability.
* **ROC-AUC Score**: The AUC score reflects the model's ability to distinguish between churners and non-churners, with higher values indicating better separability.

**Models and Performance:**

**1. Random Forest Models:**

* **Random Forest (Important Features)**: This model achieved a high **accuracy** of **92.70%** and an **AUC** score of **0.9728**, indicating strong predictive performance. The **F1 Score** of **0.93** highlights a good balance between precision and recall, making it a robust model for churn prediction.
* **Random Forest (T-Test Features)**: The model trained on statistically significant features performed similarly, with an **accuracy** of **92.57%** and an **AUC** of **0.9729**. The **F1 Score** was **0.93**, suggesting that using the t-test selected features did not significantly alter the model’s performance compared to the important features model.

**2. Logistic Regression Models:**

* **Logistic Regression (Important Features)**: The logistic regression model using important features achieved an **accuracy** of **90.59%** and an **AUC** of **0.9600**. The **F1 Score** of **0.91** was lower than that of the Random Forest models, indicating that while the logistic regression model was still effective, it did not perform as well as the ensemble-based models.
* **Logistic Regression (T-Test Features)**: This model showed a slight improvement in **accuracy** (**90.86%**) and **F1 Score** (**0.91**) over the important features model, with the **AUC** remaining at **0.9600**. However, the performance was still below that of the Random Forest models, highlighting the superiority of ensemble methods for this task.

**3. XGBoost Models:**

* **XGBoost (Important Features)**: The **XGBoost** model using important features outperformed all other models with the highest **accuracy** of **95.23%**, **precision** of **0.94**, and an **AUC** of **0.9856**. The **F1 Score** of **0.96** reflects an excellent balance between precision and recall, making this the best-performing model overall.
* **XGBoost (T-Test Features)**: The model trained on statistically significant features also performed well, with an **accuracy** of **93.62%** and an **AUC** of **0.9780**. The **F1 Score** of **0.94** indicates good performance, although slightly behind the model using important features.

A graph showing a number of different colored bars

AI-generated content may be incorrect.

**Key Insights:**

1. **XGBoost with Important Features** delivered the highest performance across all metrics. This model excelled in **accuracy**, **precision**, **recall**, **F1 Score**, and **AUC**, making it the best model for churn prediction.
2. **Random Forest** models showed strong performance with both feature sets. However, the **Random Forest (Important Features)** model slightly edged out the **Random Forest (T-Test Features)** model in terms of **accuracy** and **AUC**, although the differences were marginal.
3. **Logistic Regression** models, while efficient, did not perform as well as the ensemble-based models, especially in terms of **F1 Score** and **AUC**. The **Logistic Regression (Important Features)** and **Logistic Regression (T-Test Features)** models were more effective with the t-test features, but still lagged behind **Random Forest** and **XGBoost**.
4. **Feature Selection**: Using the **important features** yielded the best results across all models. The **T-Test Features** also performed well, particularly with **XGBoost**, but did not significantly outperform the important features.

**Conclusion:**

The **XGBoost** model using **important features** is the best choice for predicting customer churn, providing the highest **accuracy**, **precision**, **recall**, **F1 Score**, and **AUC**. The **Random Forest** model also performed well with both feature sets, showing that ensemble methods are highly effective for this task. **Logistic Regression**, while a simple and efficient model, did not match the performance of the ensemble models.

**Important Features** consistently provided better performance than **T-Test Features**, indicating that domain expertise and feature importance analysis are critical in building effective churn prediction models. The use of the right set of features plays a crucial role in improving model performance and making accurate churn predictions.