وزارة الاتصالات وتكنولوجيا المعلومات

رواد مصر الرقمية

Sales Forecasting and Demand Prediction

CLS GIZ2_AIS4_G2 Data Science

Group Members:

Dina Magdy Tolba
Dina Mohsen Amin
Fadwa Mohamed Mahmoud
Noha Mostafa Sayed
Heba Abdelsalam Abdelmotalb
Merna Ashraf Gaied

Table of Contents

1. Problem Definition

In this project, we're trying to predict whether a customer will churn after his/her subscription expires. A typical subscription lasts 30 days. The goal is to predict whether a customer will renew his/her subscription within 30 days of the previous one expiring for a music streaming service. We compared the performance of 4 machine learning models and picked the one that performed best to the

2. Dataset

The dataset used is the Customer Retention Datathon - Riyadh Edition dataset on Kaggle. The dataset contains 5 CSV files:

- **train_data.csv (946686 unique values)** - the training set that contains user id and whether they have churned or not.
- **members.csv (6769473 unique values)** - dataset with information about user members and their profile data.
- **transactions.csv 1197050 unique values)** - dataset with information about user transactions.
- **user_logs.csv (1103894 unique values)** - user behaviour logs.
- **kaggle_test_data.csv (24274 unique values)** - the test set.

3. Pre-processing

During the data preprocessing phase, we performed several essential steps to clean and prepare the data for analysis and modeling. These steps included:

3.1. Handling Missing Values:

- The **gender** column had missing values, which were filled with "**other**".
- Columns from **user_logs** such as **num_25**, **num_50**, **num_75**, **num_985**, **num_100**, **num_unq**, **total_secs**, and **active_days** had missing values and were filled with zeros.
- Any missing or invalid **bd** (age) values were replaced with the median of valid ages (between 13 and 99).

3.2. Handling Outliers:

- Users with unrealistic ages below 13 or above 99 were considered outliers and replaced with the median age.

3.4. Feature Engineering:

- User activity logs were aggregated by user (**msno**), including total counts of songs played and average daily activity.

3.4. Aggregating Log Data:

- New feature **active_days** was calculated as the number of unique days a user was active.
- An **age_bucket** feature was created using age ranges to support visualization and analysis.

3.5. Merging Data Sources:

- The main **train_data** was merged with cleaned **members**, aggregated **user_logs**, and latest transactions data using **msno** as a key.
- Only the latest transaction record per user was kept to represent recent activity.

4. Modeling

In this phase, we built and trained several machine learning models to predict whether a user will churn or not. The models were trained using the cleaned and engineered dataset from the previous preprocessing steps. The target variable was **is_churn**, a binary label indicating if the user has churned (1) or not (0).

4.1. Models Used:

- **Logistic Regression:** A simple baseline model to understand the linear relationship between features and churn.
- **Decision Tree Classifier:** A tree-based model that captures non-linear relationships and interactions between features.
- **Random Forest Classifier:** An ensemble model that uses multiple decision trees to improve performance and reduce overfitting
- **Weighted Naive Bayes:** A probabilistic model where feature contributions were adjusted using calculated weights based on feature importance

4.2. Data Splitting:

The dataset was split into training (80%) and testing (20%) sets using **train_test_split** to evaluate model performance on unseen data.

5. Model Selection:

After training the models, we evaluated their performance using multiple metrics on the test set:

5.1. Evaluation Metrics:

- **Accuracy:** Overall correctness of the model.
- **Precision**: Correctness of positive predictions.
- **Recall**: Ability of the model to identify all churned users.
- **F1 Score**: Harmonic mean of precision and recall.
- **ROC AUC**: Area under the Receiver Operating Characteristic curve.

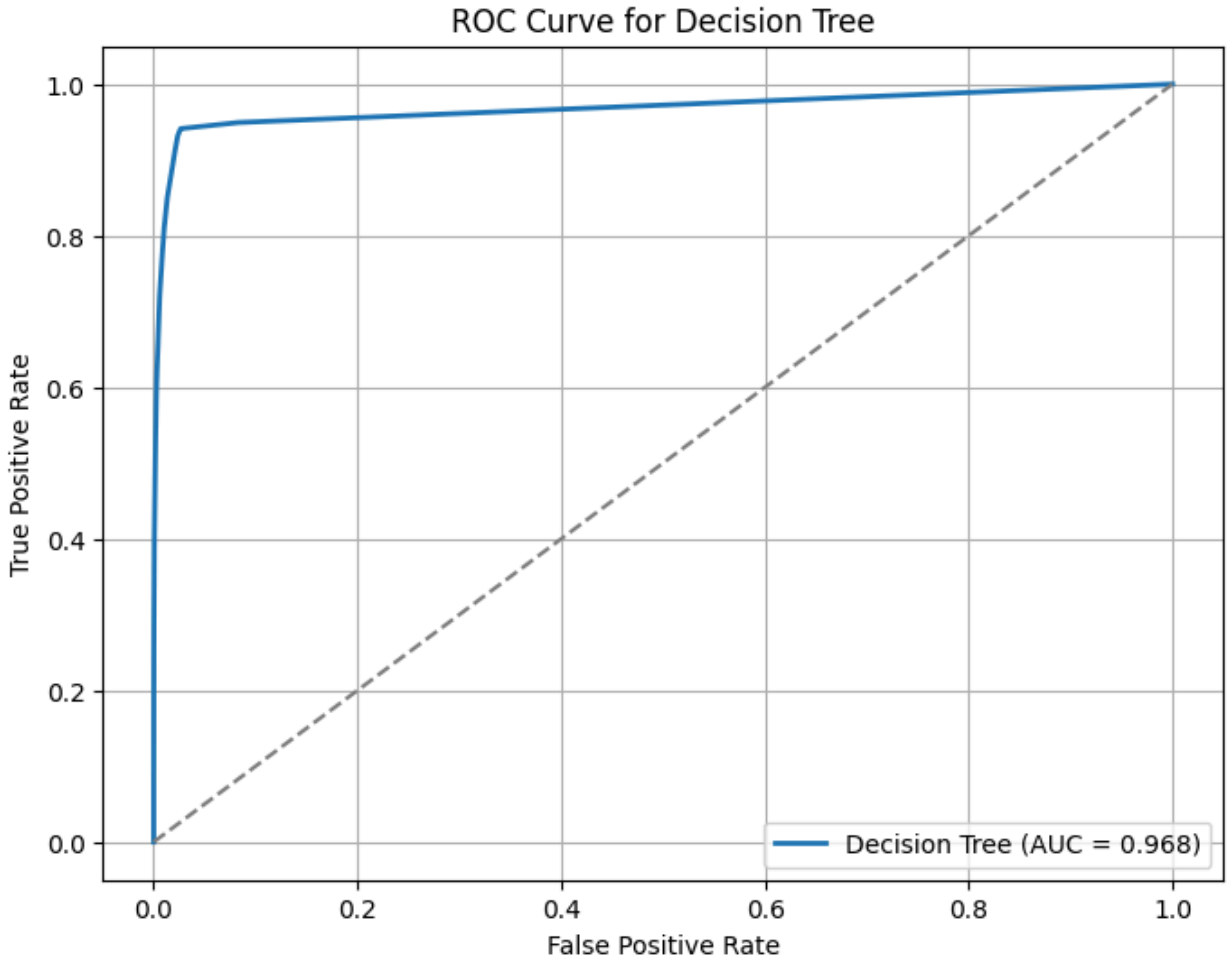
5.2. Results Comparison:

The performance metrics for the decision tree model:

```
Decision Tree AUC: 0.9682735720414651
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.97      0.98    172309
           1       0.78      0.94      0.85     17029

    accuracy                           0.97    189338
   macro avg       0.89      0.96      0.92    189338
weighted avg       0.97      0.97      0.97    189338
```

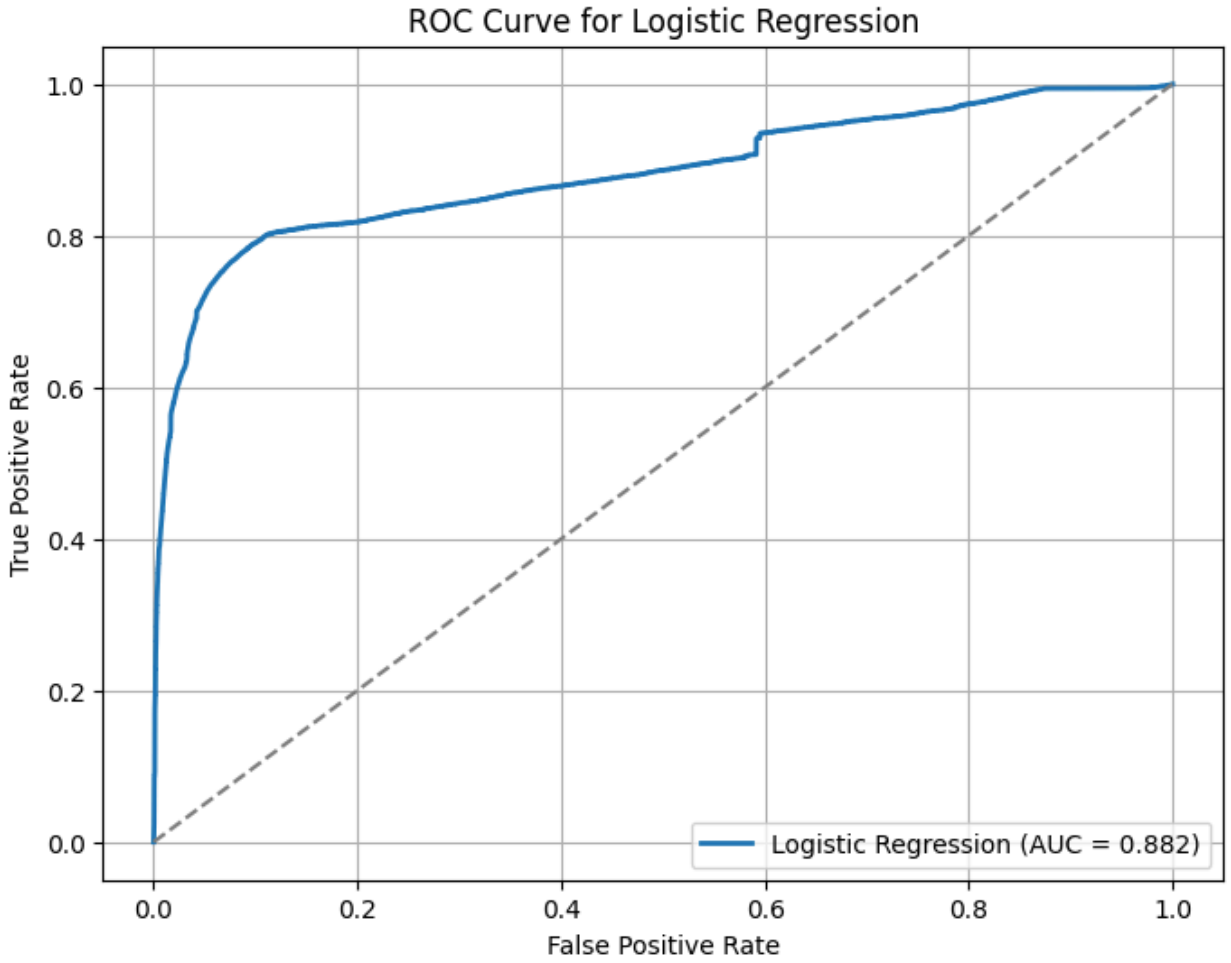


Performance metrics for the logistic regression model:

```
Logistic Regression AUC: 0.8823065401414177
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.96      0.96    172309
           1       0.62      0.69      0.65     17029

    accuracy                           0.93    189338
   macro avg       0.80      0.82      0.81    189338
weighted avg       0.94      0.93      0.94    189338
```

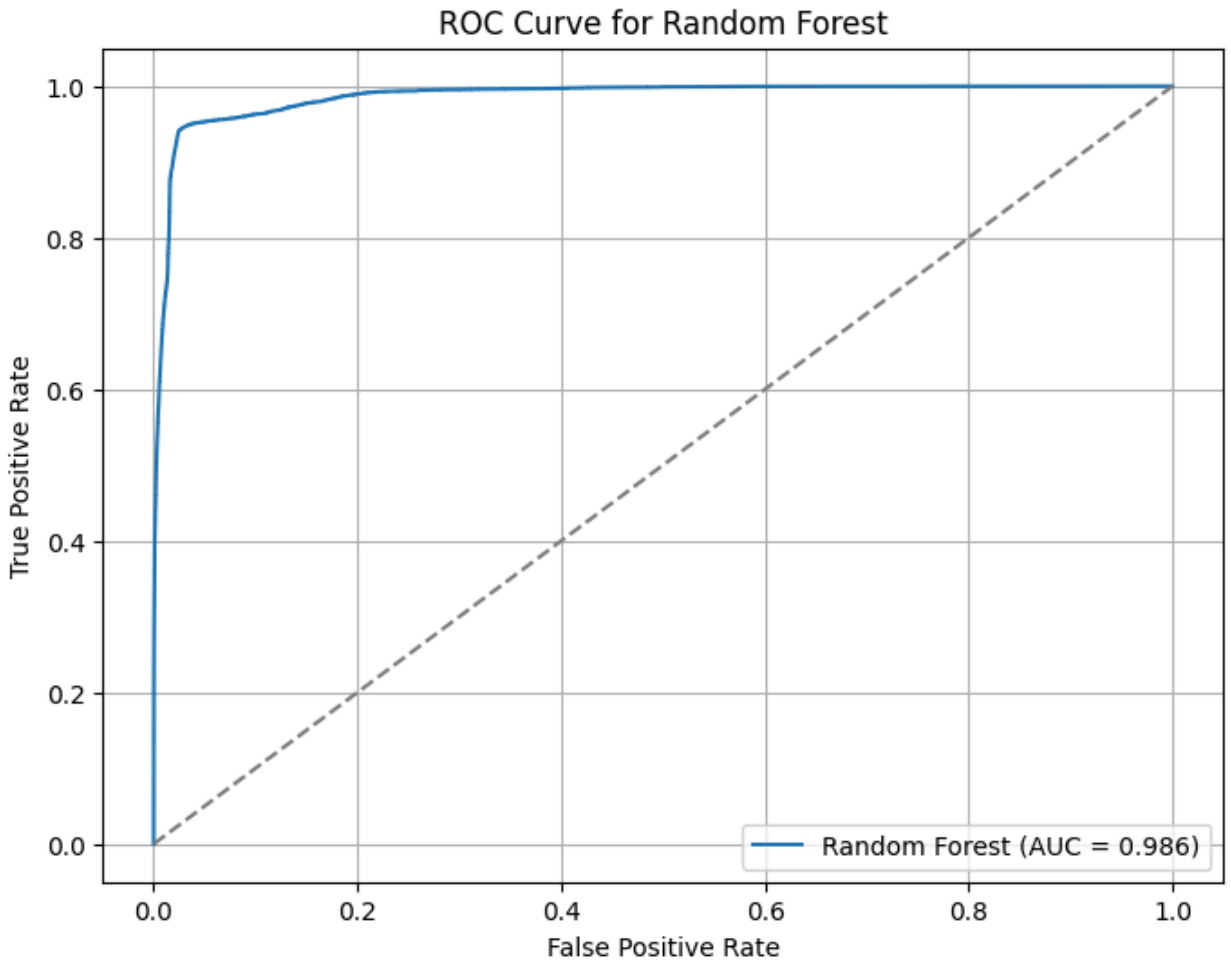


Performance metrics for random forest model:

```
Random Forest AUC: 0.9862509523604965
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.99    172309
           1       0.82      0.90      0.86     17029

    accuracy                           0.97    189338
   macro avg       0.91      0.94      0.92    189338
weighted avg       0.98      0.97      0.97    189338
```

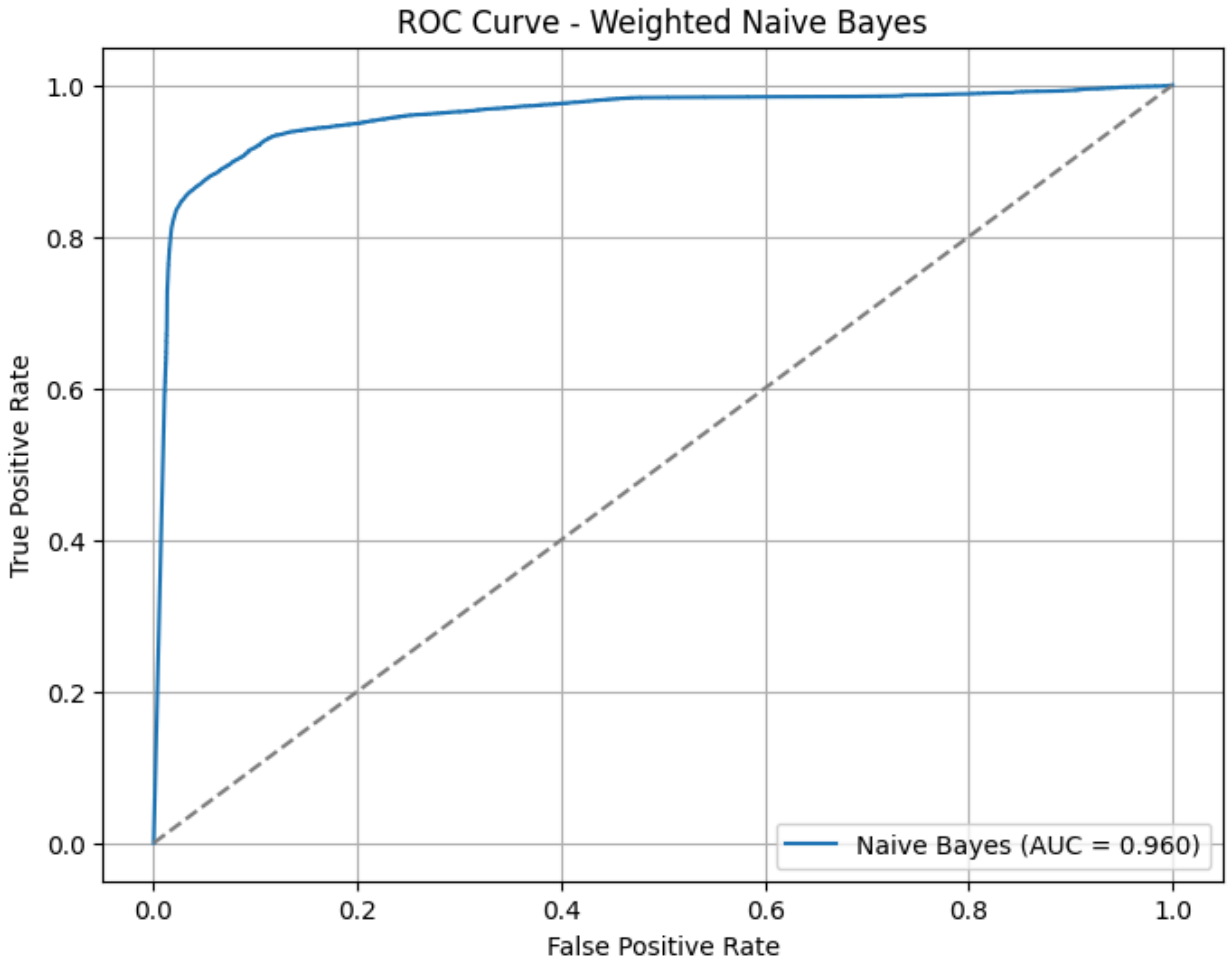


Performance metrics for the weighted naive bayes model:

```
Naive Bayes AUC: 0.9600061034132191
Naive Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98    172309
           1       0.82      0.81      0.82     17029

    accuracy                           0.97    189338
   macro avg       0.90      0.90      0.90    189338
weighted avg       0.97      0.97      0.97    189338
```



5.3. Best Model:

As seen above, all models performed really well for this use case. The difference in their scores may be negligible for the most part, however, the **Random Forest Classifier** achieved the best overall performance in terms of accuracy, F1-score, and ROC AUC. Therefore, it was selected as the final model for deployment and interpretation.

6. Model Deployment

We deployed the churn prediction model in two parts:

- **Backend (FastAPI):**
    - Built a FastAPI server that loads the trained model and provides a `/predict/` API to receive user inputs and return churn probability.
    - The server runs locally using Uvicorn and handles JSON data.
- **Frontend (Streamlit):**
    - Developed a Streamlit app with a simple form where users can enter customer features.
    - The app sends the input to the FastAPI server and immediately displays the churn prediction result.
- **Integration:**
    - Both FastAPI and Streamlit were connected to allow real-time, user-friendly churn predictions.
    - The full code and dashboard were uploaded to GitHub to prepare for future public deployment.

7. Churn Prediction Dashboard on Streamlit

We developed an interactive dashboard on Streamlit to visualize and analyze the customer churn prediction process.

The dashboard is organized into **four main tabs**:

- **1. Data Exploration:**
  Provides an overview of the dataset and key patterns in user behavior, including churn rates by different demographic and activity features.

- **2. Feature Engineering:**
  Displays insights from newly created features (such as song completion rate, membership days, discount rate, etc.) and how they impact churn behavior.

- **3. Model Comparison:**
  Presents performance comparisons between different models (Logistic Regression, Decision Tree, Random Forest, and Naive Bayes) to evaluate the best-performing approach.

- **4. Predict Churn:**
  Shows the final churn predictions made on the Kaggle test dataset and outlines potential areas for future improvements and modeling enhancements.

The dashboard enables an end-to-end understanding of the project — from raw data exploration to feature engineering, model evaluation, and final churn prediction deployment.

8. Challenges Faced and Solutions

- Data size: Initially we tried to join all the data into one large dataset to work with. We ran into performance issues when trying to process this dataset. The data also became unnecessarily large to a duplication in some record values. Instead, we worked with the data as it is and only merged dataframes whenever necessary. Another method we could've mitigated this issue is by using batch processing.

9. Future Work

First, we can test the robustness of our model on datasets from other services to further refine it. The model could also be deployed by these services to provide real time analytics using stream processing.

10. References

- https://www.kaggle.com/competitions/customer-retention-datathon-riyadh-edition/overview