Section 1:

Why are decision trees useful in customer churn prediction?

- Easy to understand: Decision trees show a clear path to making decisions, making it easy for businesses to see why a customer may leave.
- Identifies key factors: The model highlights the most important factors that influence customer churn, like low engagement or frequent complaints.
- Handles different data types: It works well with both numbers (like age, and subscription length) and categories (like payment method).
- No complex calculations needed: Unlike other models, decision trees don't require advanced math, making them easy to implement.

What business actions can be taken based on the predictions of a decision tree model?

- Prevent customer churn by offering discounts, personalized offers, or better support to customers at risk of leaving.
- Improve customer engagement by sending reminders, promotions, or special content to keep customers interested.
- Address service issues by improving customer support and fixing common problems that lead to dissatisfaction.
- Focus marketing efforts on customers who are likely to leave by offering incentives to encourage them to stay.

Section 2:

Task 1: Data Preparation and Exploration

- Load the provided dataset (customer_churn.csv) into Python:

  Images of code have been added to the project zip folder

- Perform exploratory data analysis (EDA):
  1. Display summary statistics:

     Images of code have been added to the project zip folder

  2. Identify missing values and handle them appropriately:

     Images of code have been added to the project zip folder

     There is no missing value on the dataset.

  3. Visualize data distributions using histograms and box plots:

Images of histograms and boxplots have been added to the project zip folder

4. Correlations between variables:

Images of Correlation matrix has been added to the project zip folder

The correlation matrix heatmap provides a visual representation of the relationships between different variables in your dataset. Correlation values range from -1 to 1, where:

- 1 indicates a perfect positive correlation.
- -1 indicates a perfect negative correlation.
- 0 indicates no correlation.

Here's a breakdown of the correlations between the variables based on the provided matrix:

1. **CustomerID:**
    - This variable shows very weak correlations with all other variables, which is expected since CustomerID is typically a unique identifier and not expected to have meaningful relationships with other features.
2. **Age:**
    - Age has a very weak positive correlation with Number_of_Complaints (0.06), suggesting that older customers might be slightly more likely to file complaints.
    - Age has a very weak negative correlation with Watch_Time_Hours (-0.05), indicating that older customers might watch slightly less content.
3. **Subscription_Length_Months:**
    - This variable has a very weak negative correlation with Number_of_Logins (-0.06) and Churn (-0.03), suggesting that longer subscription lengths might slightly reduce the number of logins and the likelihood of churn.
4. **Watch_Time_Hours:**
    - Watch time has a very weak negative correlation with Age (-0.05) and Churn (-0.03), indicating that customers who watch more content might be slightly younger and less likely to churn.
5. **Number_of_Logins:**
    - This variable has a very weak positive correlation with Churn (0.06), suggesting that customers who log in more frequently might be slightly more likely to churn.
6. **Payment_Issues:**

- Payment issues have a very weak positive correlation with Number_of_Complaints (0.02), indicating that customers with payment issues might be slightly more likely to file complaints.

7. **Number_of_Complaints:**
   - This variable has a very weak positive correlation with Age (0.06) and a very weak negative correlation with Resolution_Time_Days (-0.06), suggesting that older customers might file slightly more complaints, and complaints might be resolved slightly faster.
8. **Resolution_Time_Days:**
   - Resolution time has a very weak negative correlation with Churn (-0.03), indicating that faster resolution times might slightly reduce the likelihood of churn.
9. **Churn:**
   - Churn has a very weak positive correlation with Number_of_Logins (0.06) and a very weak negative correlation with Resolution_Time_Days (-0.03), suggesting that customers who log in more frequently might be slightly more likely to churn, while faster resolution times might slightly reduce churn.

Task 2: Building a Decision Tree Classifier:

- **Split the dataset into training and testing sets:**

  Images of code have been added to the project zip folder

- **Train a Decision Tree Classifier using scikit-learn:**

  Images of code have been added to the project zip folder

- **Use GridSearchCV to optimize hyperparameters (e.g., max depth, min samples split):**

  Images of code have been added to the project zip folder

- **Visualize the decision tree:**

  An image of the Decision tree has been added to the project zip folder

- **Evaluate model performance using:**
  - Accuracy: **0.615**
  - Precision: **0.306**
  - Recall: **0.358**
  - F1 Score: **0.330**

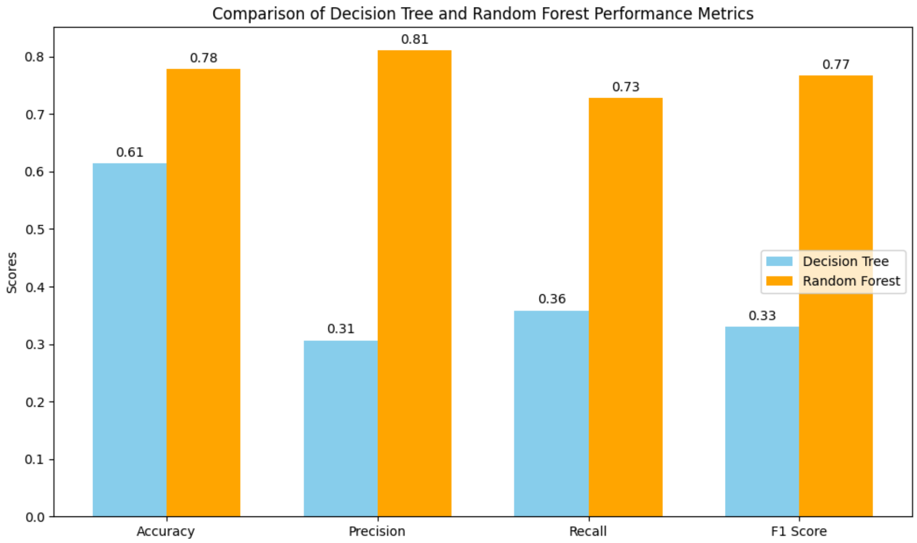
- Confusion Matrix: [[104, 43], [34, 19]]

Task 3: Improving Performance with Random Forests

- **Train a Random Forest Classifier:**

  Images of code have been added to the project zip folder

- **Compare its performance against the Decision Tree model:**



- **Analyze feature importance:**

  Images of feature importance for both the Decision tree and Random forest have been added to the project zip folder

1. Top Important Features:
   - **Watch_Time_Hours**
     - Highest importance in both Decision Tree and Random Forest.
     - Indicates that user engagement is a major predictor.
   - **Number_of_Logins**
     - Highly important in both models.
     - Suggests frequent logins correlate with key predictions.
   - **Subscription_Length_Months**
     - Medium in Decision Tree, higher in Random Forest.
     - Long-term subscriptions play a stronger role in Random Forest.

2. Medium Important Features:

   - **Resolution_Time_Days**
     - Medium importance in both Decision Tree and Random Forest.
     - Faster support resolution correlates with customer retention.
   - **Age**
     - High in Decision Tree, medium in Random Forest.
     - DT emphasizes user demographics more.
   - **Number_of_Complaints**
     - Medium in both models.
     - Customer dissatisfaction is a significant predictor.
   - **Membership_Type (Standard/Premium)**
     - Medium in both models.
     - Premium members contribute slightly more to predictions.

3. Least important Features:

   - **Preferred_Content_Type (Sports, TV Shows, Movies)**
     - Low in both Decision Tree and Random Forest.
     - Suggests content choice is less influential than engagement metrics.
   - **Payment_Method (Credit Card, PayPal, Others)**
     - Low importance in both models.
     - Implies that all payment methods behave similarly.
   - **Payment_Issues**
     - Lowest importance in both models.
     - Shows that billing problems have little impact on predictions.

- **Why the Random Forest model performed better:**

  1. Ensemble Learning Advantage
  - Random Forest is an ensemble learning method that builds multiple decision trees and aggregates their predictions (via voting for classification). This approach reduces overfitting compared to a single Decision Tree.
  - A single Decision Tree can easily overfit the training data, especially when it grows deep or has complex splits. Random Forest mitigates this by averaging predictions across many trees, which smooths out noise and reduces variance.

  2. Feature Randomness
  - In addition to using bootstrapped samples of the data, each tree in a Random Forest considers only a random subset of features at each split. This introduces diversity among the trees, reducing the correlation between them.
  - By decorating the trees, Random Forest avoids relying too heavily on specific features, which can happen in a single Decision Tree if certain features dominate the splits.

3. Handling Class Imbalance with SMOTE

- Both models used **SMOTE** to address class imbalance in the dataset. However, Random Forest inherently handles imbalanced datasets better than a single Decision Tree because of its ensemble nature.
- When combined with SMOTE, Random Forest benefits even more because the resampled data provides balanced training examples for each tree in the forest.

4. Hyperparameter Tuning

- Both models underwent hyperparameter tuning using **GridSearchCV** , but Random Forest has more parameters to optimize (e.g., n_estimators, max_depth, min_samples_split, class_weight), allowing it to achieve a better configuration.

5. Robustness to Noise

- Single Decision Trees are highly sensitive to small changes in the data, such as noise or outliers, which can lead to poor generalization. Random Forest, by aggregating predictions from multiple trees, is less affected by such issues.
- This robustness is particularly important in real-world datasets like this one, where there may be noisy or irrelevant features.

6. Bias-Variance Tradeoff

- A single Decision Tree often suffers from high variance, meaning it performs well on training data but poorly on test data. Random Forest reduces variance by averaging predictions, achieving a better bias-variance tradeoff.
- The reduction in variance improves the model's ability to generalize to new data, as seen in the Random Forest's higher test set performance.

7. Scalability and Feature Importance

- Random Forest can handle a large number of features and interactions between them more effectively than a single Decision Tree. It also provides a measure of feature importance, which can help identify key predictors of churn.
- While the Decision Tree might focus too much on a few dominant features, Random Forest distributes importance across multiple features, capturing more complex relationships.

Task 4: Business Insights and Recommendations

We created this diagram using the Decision Tree model to help us determine the features that contribute the most to the feature Churn. (The diagram can be found in the collab document)

1. **Watch Time Hours** – Lower watch time signals disengagement and higher churn risk.
2. **Logins** – Infrequent logins suggest low engagement and potential churn.
3. **Support Resolution Time** – Delayed issue resolution increases churn likelihood.
4. **Subscription Length** – Newer users churn more, indicating retention challenges.
5. **Age** – Certain age groups are more prone to churn; targeted marketing is needed.
6. **Complaints** – More complaints correlate with higher churn risk.
7. **Membership Type** – Premium users churn less; lower-tier plans offer less perceived value.

What are some practical ideas that can help reduce customer churn?

- **Boost User Engagement**
  Low watch time is a strong churn predictor. StreamFlex can leverage AI-driven recommendations to personalize content suggestions based on user preferences, viewing history, and trending topics. Sending notifications or emails about new or unfinished series can encourage users to return. Additionally, well-structured playlists and an intuitive "continue watching" feature will enhance accessibility and engagement.
- **Improve Customer Support**
  Slow issue resolution frustrates users and increases churn. Implementing AI-powered chatbots can provide instant responses to common queries, such as billing or account settings. Prioritizing high-tier subscribers and long-term users for faster support can improve loyalty. Offering self-service options and real-time chat support in a dedicated help center will further enhance the customer experience.
- **Strengthen Onboarding for New Users**
  New users are the most likely to churn, making an interactive onboarding process essential. A guided platform tour highlighting key features, personalized content suggestions, and short tutorials can help users find value quickly. Sending "top picks for you" within the first few weeks can establish viewing habits. Providing limited-time access to premium features or exclusive content can also encourage deeper engagement and long-term retention.

Business Strategies

1. **Improve Personalized Recommendations**
- **Why?** Low watch time leads to higher cancellations. Users need quick access to relevant content.
- **How?** StreamFlex will enhance recommendations using AI based on viewing history and preferences. Weekly personalized emails and a **"Continue Watching"** section will encourage engagement.

2. **Enhance Customer Support Efficiency**
- **Why?** Slow issue resolution frustrates customers, increasing churn.
- **How?** AI chatbots will handle common issues instantly, while live chat will assist with complex cases. StreamFlex will aim to resolve most requests within **24 hours** to boost customer satisfaction.

3. **Offer Loyalty Programs**
- **Why?** Short-term subscribers are more likely to cancel, but incentives can improve retention.
- **How?** Discounts for long-term plans and exclusive perks (early content access, reward-based "watch series" programs) will encourage continued engagement and commitment.