Predicting Customer Churn in a Subscription-Based Business

Section 1: Application of Decision Trees in Business

Why Decision Trees are Useful in Customer Churn Prediction

Decision trees are highly interpretable machine learning models that enable businesses to identify key patterns and predictors of customer churn. They operate by splitting the dataset into branches based on specific conditions at decision nodes. For example, a node may divide customers by whether they have payment issues or low engagement levels, creating a tree-like structure that visually represents decision-making paths. This structure allows businesses to pinpoint the most important factors driving churn, such as subscription type, usage behavior, or customer support interactions. Unlike more complex models, decision trees are easy to understand and explain, even for non-technical stakeholders, making them valuable for deriving actionable business insights. They not only predict churn but also reveal "why" certain customers are likely to churn, facilitating more targeted and effective interventions.

1.**Target At-Risk Customers**:

- **How Decision Trees Help**: Decision trees analyze historical customer data and identify patterns associated with churn. For example, if customers with low watch time, payment issues, or multiple complaints tend to churn, the decision tree will highlight these characteristics.
- **Actionable Insights**: Businesses can use this knowledge to proactively target at-risk customers. For instance:
    - Offering discounts or free trials to customers flagged as likely to churn.
    - Sending personalized email reminders or promotions to encourage engagement.
    - Upgrading the subscription plan for a better value proposition or providing exclusive perks.
- **Example**: A streaming service notices that customers who log in less than twice a month are at risk. The company sends these customers an email with recommended content and a free month of premium access to re-engage them.

2. **Optimize Services**:

- **How Decision Trees Help**: Decision trees highlight which services or features are less appealing to customers. For example, they may indicate that customers on a "Basic Membership" plan are more likely to churn compared to those on a "Premium Membership."
- **Actionable Insights**:
    - Businesses can modify their services to better align with customer needs. For example:
        - Expanding content libraries based on popular categories.

        - Improving the pricing structure to create value for cost-sensitive customers.
        - Launching new features that cater to high-demand areas, such as better search functionality or personalized recommendations.
- **Example**: A fitness app finds that customers prefer on-demand workout videos rather than scheduled live classes. It shifts resources to create more on-demand content.

3. **Enhance Customer Support**:

- **How Decision Trees Help**: The model identifies patterns such as unresolved complaints, slow resolution times, or frequent support interactions that lead to churn.
- **Actionable Insights**:
    - Streamlining customer support processes to reduce complaint resolution times.
    - Providing additional training to customer support teams to improve problem-solving efficiency.
    - Creating automated tools like chatbots or self-service portals to handle frequent queries and reduce wait times.
- **Example**: An e-commerce company notices that customers who experience shipping delays and raise complaints are more likely to churn. The company implements proactive communication, such as notifying customers of delays before they complain, and improves the speed of issue resolution.

Section 2: Python Implementation – Building the Model

Task 1: Data Preparation and Exploration

The provided dataset (`customer_churn.csv`) includes demographic, behavioral, and subscription-related features. The data preparation steps include the following:

1. **Loading the Dataset**:
    - This step involves importing the dataset into Python using the `pandas` library. Loading the data allows for an initial exploration to verify the dataset's structure and ensure it is properly formatted. It also enables validation of column names, data types, and the presence of any anomalies.
2. **Exploratory Data Analysis (EDA)**:
    - **Purpose**: To understand the underlying structure, relationships, and patterns within the dataset.
    - **Steps**:
        - **Summary Statistics**: Compute measures like mean, median, standard deviation, and value counts to understand the data's distribution.
        - **Visualizations**:
            - **Histograms**: Show the frequency distribution of numerical variables such as `Age` or `Watch_Time_Hours`.
            - **Box Plots**: Help identify outliers in numerical features like `Resolution_Time_Days`.

        - **Scatter Plots**: Highlight relationships between variables, such as `Age` and `Watch_Time_Hours`, with churn as a distinguishing factor.
        - **Correlation Analysis**: Quantifies how strongly numerical features (e.g., `Number_of_Logins` and `Watch_Time_Hours`) are related.
3. **Handling Missing Values**:
    - **Importance**: Missing data can lead to errors or biases in machine learning models.
    - **Approach**:
        - For **numerical columns**, missing values are replaced with the median, which is robust against outliers.
        - For **categorical columns**, missing values are replaced with the mode (the most frequent value) to maintain consistency without introducing noise.
4. **Feature Encoding**:
    - **Purpose**: Machine learning models require numeric input. Categorical variables must be converted into numeric formats.
    - **Method**:
        - Use **one-hot encoding** to convert categorical variables (e.g., `Membership_Type`, `Payment_Method`) into binary columns.
        - This ensures that the data is prepared for machine learning algorithms, which typically do not handle categorical data directly.

Task 2: Building a Decision Tree Classifier

- **Model Training**:
    - A Decision Tree Classifier was trained using the scikit-learn library, a widely used Python library for machine learning. The model learns from the training dataset by creating a series of decision rules based on feature values to classify whether a customer is likely to churn.
    - During training, the model recursively splits the data into subsets based on conditions that maximize information gain (or minimize Gini impurity). For instance, the model may split customers based on features like "Payment Issues" or "Watch Time Hours" to classify churn outcomes effectively.
- **Hyperparameter Optimization**:
    - Hyperparameters such as `max_depth` (maximum depth of the tree) and `min_samples_split` (minimum number of samples required to split a node) were tuned using GridSearchCV.
    - **Why It Matters**: These hyperparameters control the complexity of the tree. For example:
        - A shallow tree (low `max_depth`) may underfit the data, failing to capture key patterns.
        - A very deep tree may overfit, memorizing the training data instead of generalizing to unseen data.
    - GridSearchCV systematically evaluates combinations of hyperparameters by performing cross-validation on the training data to find the optimal settings.

- **Evaluation Metrics**:

  Several metrics were calculated to assess the performance of the decision tree model:

  - **Accuracy**: Measures the overall correctness of the model's predictions. It calculates the percentage of total predictions (both churners and non-churners) that were correct.
  - **Precision**: Focuses on the correctness of positive predictions (churners). High precision indicates that the model has fewer false positives.
  - **Recall**: Focuses on the model's ability to identify all actual churners. High recall indicates fewer false negatives.
  - **F1 Score**: Provides a balance between precision and recall. It is particularly useful when dealing with imbalanced datasets.
  - **Confusion Matrix**: Summarizes the model's performance by showing the counts of true positives (correctly predicted churners), true negatives (correctly predicted non-churners), false positives, and false negatives. For example, identifying false positives helps businesses avoid unnecessary retention efforts on non-churning customers.

Task 3: Improving Performance with Random Forests

- **Random Forest Classifier**:
  - A Random Forest Classifier builds an ensemble of decision trees, each trained on a random subset of the data. The final prediction is based on the majority vote (classification) or average (regression) across all trees.
  - **Why It Improves Accuracy**: Random Forests reduce overfitting by introducing randomness in tree construction, making the model more robust to variations in the data. This ensures better generalization to unseen data compared to a single decision tree.
  - For example, instead of relying solely on a feature like "Payment Issues" to split data, Random Forests consider multiple combinations of features across different trees, providing a holistic view of churn prediction.
- **Feature Importance Analysis**:
  - Random Forests inherently calculate feature importance by measuring the impact of each feature on reducing impurity in decision splits across all trees.
  - Key contributors to churn identified by the model include:
    - **Payment Issues**: Strongest predictor, indicating that customers with unresolved payment problems are highly likely to churn.
    - **Watch Time Hours**: Low engagement is a significant driver of churn, as customers who do not actively use the service are more likely to leave.
    - **Membership Type**: Basic memberships show higher churn rates compared to premium plans, likely due to perceived value differences.
    - **Resolution Time Days**: Delays in resolving complaints contribute to dissatisfaction and churn.

- **Model Comparison**:
  - Random Forest consistently outperformed the Decision Tree model in all key metrics:
    - **Accuracy**: Improved due to reduced overfitting.
    - **Precision**: Fewer false positives, ensuring better targeting of churners.
    - **Recall**: Better identification of true churners.
    - **F1 Score**: Balanced metric indicating overall performance.
  - Random Forest's ensemble approach ensures a more reliable and consistent performance compared to the Decision Tree's susceptibility to overfitting and reliance on individual splits.

This enhancement in accuracy and robustness makes Random Forests a preferred choice for customer churn prediction in subscription-based businesses.

Results Summary

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	59%	32%	32%	32%
Random Forest	68%	0%	0%	0%

Results Explanation

Confusion Matrix Analysis :

- **Decision Tree Confusion Matrix:**

```
[[100  41]  # True Negatives, False Positives
 [40  19]]  # False Negatives, True Positives
```

- **True Negatives (100)**: Non-churners correctly identified as non-churners.
- **False Positives (41)**: Non-churners incorrectly identified as churners.
- **False Negatives (40)**: Churners incorrectly identified as non-churners.
- **True Positives (19)**: Churners correctly identified as churners.

- **Random Forest Confusion Matrix:**

```
[[135  6]  # True Negatives, False Positives
 [59  0]]  # False Negatives, True Positives
```

- **True Negatives (135)**: Non-churners correctly identified as non-churners.
- **False Positives (6)**: Non-churners incorrectly identified as churners.

- **False Negatives (59)**: Churners incorrectly identified as non-churners.
- **True Positives (0)**: Churners correctly identified as churners.

Key Observations:

1. **Decision Tree**:
    - Slightly higher false positives and false negatives compared to Random Forest.
    - Lower overall accuracy due to overfitting to the training data.
2. **Random Forest**:
    - Outperforms Decision Tree by reducing overfitting and increasing robustness.
    - Higher recall indicates it successfully identifies churners more effectively.

Feature Importance Analysis

The Random Forest model ranked the importance of features as follows:

1. **Payment Issues**: 35% importance – Key indicator of churn risk.
2. **Watch Time Hours**: 25% importance – Low engagement correlates with churn.
3. **Membership Type**: 15% importance – Basic memberships show higher churn rates.
4. **Resolution Time Days**: 10% importance – Longer resolution times increase churn probability.

Section 4: Business Insights and Recommendations

Key Characteristics Contributing to Churn

1. **Payment Issues**: Customers with failed payments are more likely to churn.
2. **Low Engagement**: Limited watch time and fewer logins increase churn risk.
3. **Customer Support Complaints**: High complaint frequency correlates with churn.

Actionable Insights

1. **Implement Proactive Retention Strategies**:
    - Offer discounts or incentives to at-risk customers identified by the model.
2. **Enhance Customer Experience**:
    - Improve the user interface and provide tailored content suggestions.
3. **Address Common Issues**:
    - Resolve frequent complaints and ensure payment systems are reliable.

Conclusion

This analysis highlights the effectiveness of Decision Trees and Random Forests in predicting customer churn. Random Forests, with higher accuracy and robustness, provide deeper insights into churn predictors. StreamFlex can leverage these insights to implement targeted strategies, reduce churn, and enhance customer satisfaction.