Visualisation and Model Development

Name: Luong Hai Chau

Student I.D: A00117495

Subject Code: BDA601

Subject Name: Visualisation and Model Development

Assessment No.: 2

Title of Assessment: Design

Lecturer: Sheng Shen

Date: Nov 2024

Table of Contents

Introduction

Customer churn represents a significant challenge for businesses, particularly in highly competitive industries like retail. For **Big Retail**, an online retail company, declining visitor numbers and low conversion rates have made it critical to understand and address customer churn. Churn leads to revenue loss, increased customer acquisition costs, and impacts on business sustainability. Addressing this issue is crucial for enhancing profitability and maintaining market share.

This report aims to develop a data-driven approach to predict customer churn using machine learning techniques. By identifying at-risk customers and understanding key drivers of churn, Big Retail can develop targeted strategies to improve customer retention. The approach involves data preprocessing, model training and evaluation, feature importance analysis, and strategic recommendations for business improvements.

Model Development

Data preprocessing

Data preprocessing is a crucial step to ensure the dataset is clean, consistent, and ready for machine learning algorithms. Following recommendations of **Hidayat, A. (2024)**, this involved:

- Handling Missing Value
- Encoding Categorical Variables
- Scaling Numerical Features

Handling Missing Value

The common challenge of using consumer behavioural data is missing data (Awan et al., 2021 cited in **Liu, Y. et al, 2024**). This might lead to invalid conclusions (Zhang et al, 2022 cited in **Liu, Y. et al, 2024)** and the handling approach can significantly affect the outcomes of the analysis. Regarding the research of Tiwaskar, S. et al. 2024, imputation is a robust approach.

In this project, the '**TotalCharges**' column had missing values that needed to be addressed by employing **Mean Imputation**. This approach is chosen as it maintains the overall data distribution. While mean imputation is simple and effective, this method can distort data distribution and underestimate variability **(Khan, M. A. 2024**). In future implementations, more sophisticated methods such as **k-Nearest Neighbors imputation** or **multiple imputations** could be explored to assess if the missing values affect the model's results.

```
# Handle missing values
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
imputer = SimpleImputer(strategy='mean')
df['TotalCharges'] = imputer.fit_transform(df[['TotalCharges']])
```

```
Missing values before imputation: 11
Missing values after imputation: 0
Training set shape: (5634, 15)
Testing set shape: (1409, 15)
```

Encoding Categorical Variables

Categorical features are converted into numerical format, necessary for model processing (Zhu, W., Qiu, R., & Fu, Y. 2024). The **Label Encoding** approach is leveraged to encode the original label output space into a new label space as **Tang, J. et al 2024** recommended. Employing Label Encoder to convert categorical data into numerical data that makes the data more suitable for algorithm processing (**Udandarao, V., & Gupta, P. 2024**)

```
# Encode categorical variables
for col in categorical_columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col].astype(str))
```

Scaling Numerical Features

According to **Rashmi, C. R., & Shantala, C. P. (2024)**, feature scaling is a significant step in data preprocessing to ensure the performance of machine learning models, and consistent feature scales. In their research, **StandardScaler** is introduced as one of the most popular approaches, supporting in normalizing the features distribution.

```
# Scale numerical features
scaler = StandardScaler()
df[numerical_columns] = scaler.fit_transform(df[numerical_columns])
```

Model Selection & Training

In the context of customer churn analysis for Big Retail, the model is designed to accurately predict whether a customer is likely to churn based on various input features, including contract type, tenure, monthly charges, and other relevant factors. Research by **De Caigny, A. et al. (2018)** highlights **Decision Trees** as a popular and effective algorithm for customer churn prediction due to their strong predictive performance. Additionally, **Kim, S., & Lee, H (2022)** emphasize that the tree-structured model is particularly effective for churn prediction in influencer commerce, further supporting its suitability for this analysis.

Building upon the strengths of **Decision Trees** and **Random Forest** algorithms are also utilized to enhance prediction accuracy and model robustness. Random Forest constructs multiple **Decision Trees** and aggregates their outputs to make more reliable and stable predictions. This approach reduces the risk of overfitting and provides a higher level of predictive performance, making it well-suited for complex datasets. This algorithm is leveraged to build the model, offering insight of utilizing various approaches in constructing an accurate predictive model.

```
# Decision Tree
dt_model = DecisionTreeClassifier(random_state=42)
dt_params = {
    'max_depth': [3, 5, 7, 9],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Random Forest
rf_model = RandomForestClassifier(random_state=42)
rf_params = {
    'n_estimators': [100, 200],
    'max_depth': [3, 5, 7],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
# Models
models =[("Decision Tree", dt_model, dt_params), ("Random Forest", rf_model, rf_params)]
training_results = []
for name,model, params in models:
    print(f"\nTraining and Evaluating {name} Model:")
    best_model, feature_importance, results = train_and_evaluate_model(
        model, params, X_train, X_test, y_train, y_test
    )
    training_results.append((name, best_model, feature_importance, results))
```

Additionally, selecting the right hyperparameters is crucial for optimizing model performance. **Rasheed, S. et al, (2024)** recommend using **GridSearchCV** to identify the combination of parameters for the model due to the popularity of this technique in machine learning in terms of its ability to fine-tune models.

```
# Hyperparameter tuning with GridSearchCV
grid_search = GridSearchCV(model, params, cv=5, scoring='roc_auc', n_jobs=-1)
grid_search.fit(X_train, y_train)

print("\nBest Parameters:", grid_search.best_params_)
print("Best Cross-Validation Score:", grid_search.best_score_)
best_model = grid_search.best_estimator_
```

Model Evaluation

Model evaluation plays a crucial role in assessing the performance and reliability of a predictive model. According to **Vujovic, Z. (2021)**, the performance can be measured using several key metrics:

- **Accuracy:** Measures the proportion of correct predictions out of the total number of predictions.
- **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):** evaluates the model's ability to discriminate between the classes (C**arrington, A. M. et al, 2023**). A higher AUC-ROC score indicates better performance in distinguishing between customers who are likely to churn and those who will remain.
- **Average Precision** summarizes the precision-recall curve, which is particularly useful for evaluating models on imbalanced datasets.
- **Confusion Matrix** provides a detailed breakdown of the model's performance by showing true positives, true negatives, false positives, and false negatives. This helps identify where the model might be making errors
- **Classification Report** gives additional metrics such as precision, recall, and F1-score for both classes, offering a more in-depth performance analysis.

Code

```
# Model evaluation
y_pred = best_model.predict(X_test)
y_pred_proba = best_model.predict_proba(X_test)[:, 1]

# Collect evaluation metrics
results = {
    'Accuracy': accuracy_score(y_test, y_pred),
    'AUC-ROC': roc_auc_score(y_test, y_pred_proba),
    'Average Precision': average_precision_score(y_test, y_pred_proba),
    'Confusion Matrix': confusion_matrix(y_test, y_pred),
    'Classification Report': classification_report(y_test, y_pred)
}
```

Results:

```
Training and Evaluating Decision Tree Model:

Best Parameters: {'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 10}
Best Cross-Validation Score: 0.8188259432439778

Model Evaluation Results:
Accuracy: 0.773
AUC-ROC: 0.810
Average Precision: 0.547

Confusion Matrix:
 [[947  88]
 [232 142]]

Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.91      0.86      1035
           1       0.62      0.38      0.47       374

    accuracy                           0.77      1409
   macro avg       0.71      0.65      0.66      1409
weighted avg       0.75      0.77      0.75      1409
```

```
Training and Evaluating Random Forest Model:

Best Parameters: {'max_depth': 7, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 100}
Best Cross-Validation Score: 0.8371688254709424

Model Evaluation Results:
Accuracy: 0.783
AUC-ROC: 0.834
Average Precision: 0.645

Confusion Matrix:
 [[945  90]
 [216 158]]

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.91      0.86      1035
           1       0.64      0.42      0.51       374

    accuracy                           0.78      1409
   macro avg       0.73      0.67      0.68      1409
weighted avg       0.77      0.78      0.77      1409
```

Interpretation of Churn Analysis

Model Performance

Decision Tree Model

- **Metrics**:
  - **Accuracy**: 77.3%
  - **AUC-ROC**: 0.810
  - **Average Precision**: 0.547
- **Confusion Matrix**:
  - True Negatives: 947, False Positives: 88
  - False Negatives: 232, True Positives: 142
- **Classification Report**:
  - **Non-Churn (0)**: Precision 0.80, Recall 0.91, F1-score 0.86
  - **Churn (1)**: Precision 0.62, Recall 0.38, F1-score 0.47

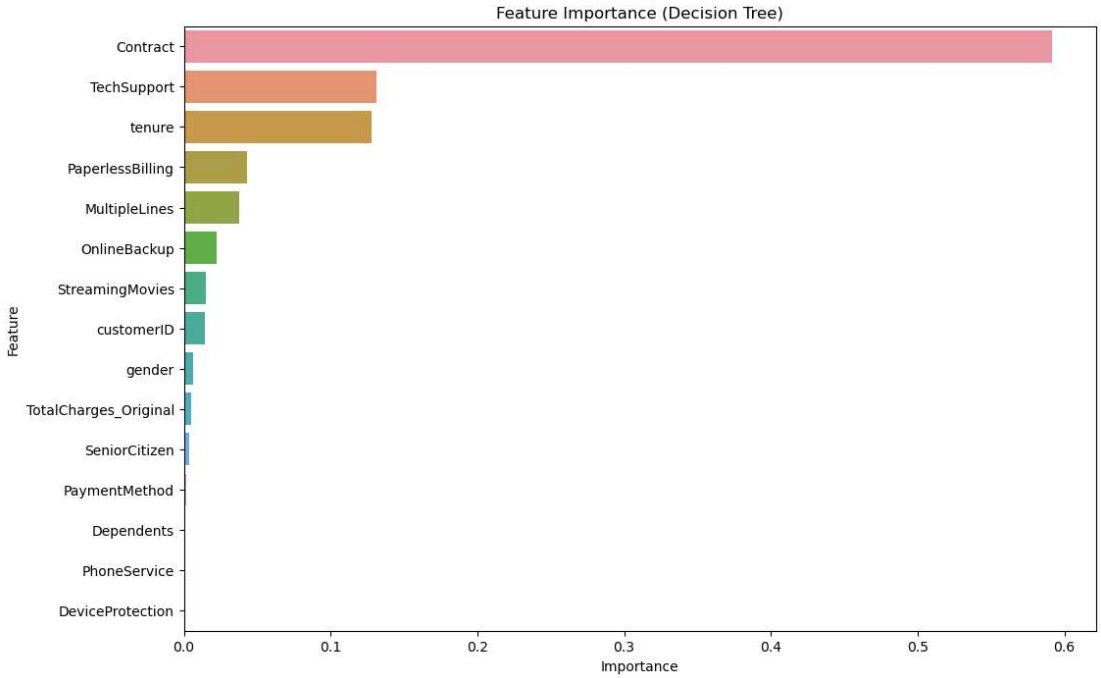
Random Forest Model

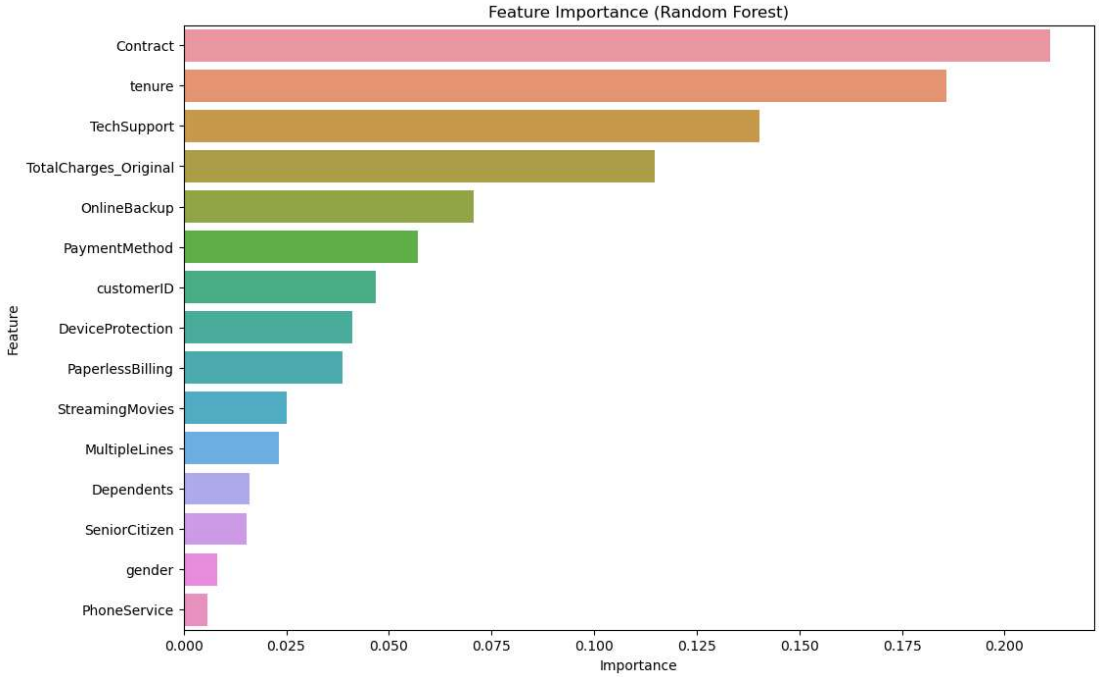
- **Metrics**:
  - **Accuracy**: 78.3%
  - **AUC-ROC**: 0.834
  - **Average Precision**: 0.645
- **Confusion Matrix**:
  - True Negatives: 945, False Positives: 90
  - False Negatives: 216, True Positives: 158
- **Classification Report**:
  - **Non-Churn (0)**: Precision 0.81, Recall 0.91, F1-score 0.86
  - **Churn (1)**: Precision 0.64, Recall 0.42, F1-score 0.51

The models demonstrate strong performance, with accuracy exceeding **77%** and AUC-ROC scores above **0.8**. These metrics indicate a reliable ability to predict customer churn, enabling effective identification of high-risk customers and facilitating targeted retention strategies.

Feature Importance

In the research by **Ehsani, F., & Hosseini, M. (2024)**, feature importance is a technique used to score the input features to identify which attributes hold the greatest influence in predicting customer churn. In the context of Big Retail, the feature importance scores from both the Decision Tree and Random Forest models are illustrated in the following diagrams.

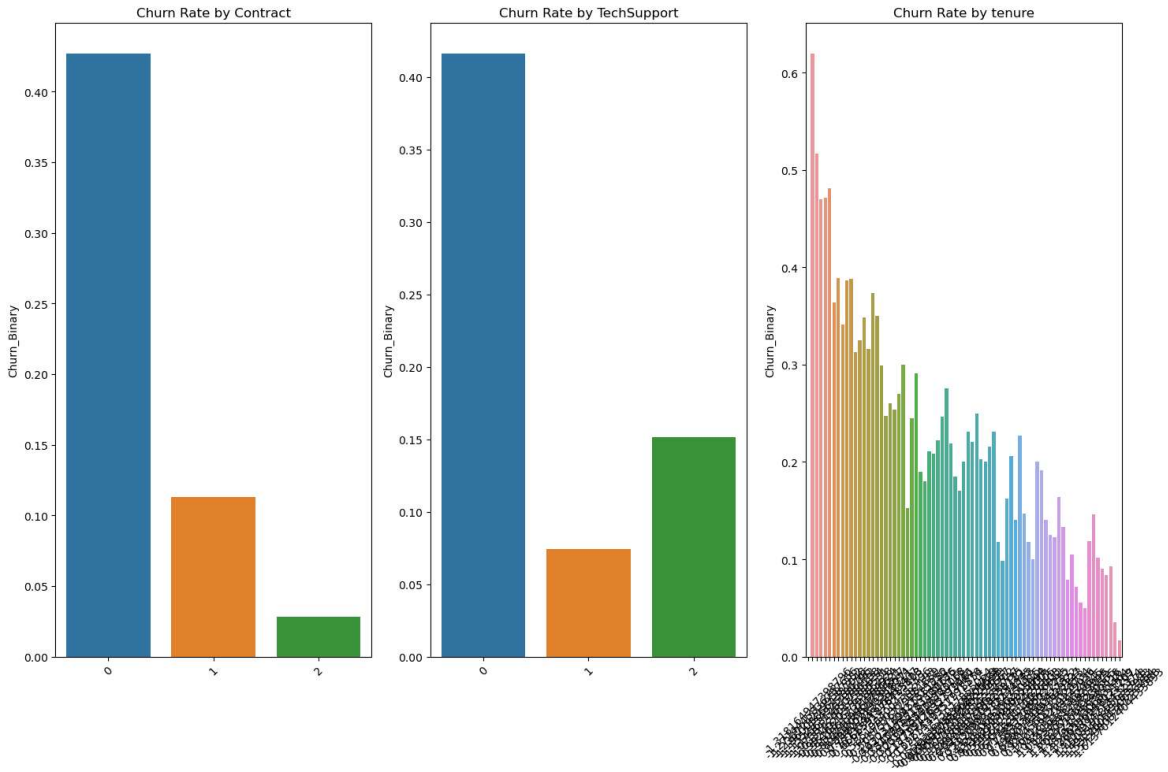




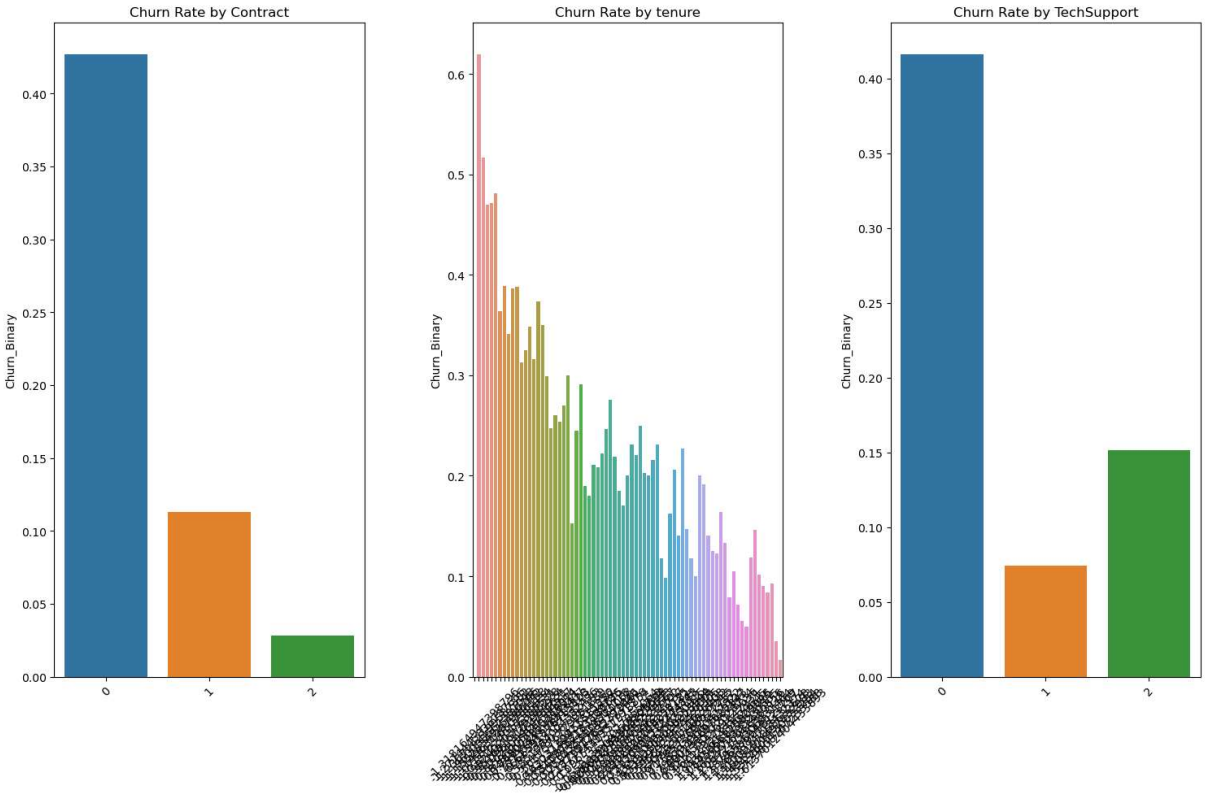
The following top features are identified in terms of results that play a crucial role in predicting customer churn:

- **Contract:** In both models, Contract feature is ranked as the most important feature, indicating it has a significant impact on predicting whether a customer will churn.
- **Tech Support:** indicates whether a customer has opted for technical support services. The analysis shows that customers who do not have access to tech support are more likely to churn. Both models rank Tech Support as a highly influential factor, suggesting that the absence of support services negatively affects customer satisfaction and increases the likelihood of churn.
- **Tenure:** refers to how long a customer has been with the company. It is the second most important feature in both models, highlighting the strong relationship between customer loyalty (tenure) and churn risk. Generally, customers with shorter tenures are more likely to churn, while those with longer tenures tend to remain loyal.

Analyzing Churn Pattern



Churn Rates of Decision Tree



Churn Rates of Random Forest

The churn analysis across both the Decision Tree and Random Forest models reveals key features that significantly impact customer churn: Contract Type, Tenure, and Tech Support.

- **Contract Type:** The analysis shows that customers with **month-to-month contracts** (labelled as **0**) have the highest churn rate. In contrast, the churn rate declines significantly for customers with **one-year contracts** (labelled as **1**) and is lowest for those with **two-year contracts** (labelled as **2**). This trend suggests that long-term contracts foster customer loyalty.
- **Tenure:** Customers with **shorter tenure** (newer customers) exhibit a significantly higher churn rate, particularly in the initial months. The data shows a steady decline in churn as tenure increases, with long-term customers being less likely to leave.
- **Tech Support**: The absence of **technical support services** is associated with the highest churn rate. Customers who lack tech support (labelled as **0**) are more prone to leaving, whereas those with some level of support (labelled as **1**) have a lower churn rate. Interestingly, the churn rate slightly increases for customers with comprehensive tech support (labelled as **2**), suggesting potential gaps in service quality or unmet expectations

In terms of insights from the churn analysis, the **potential cost savings for Big Retail** can be estimated using the following formula:

$$avg\_customer\_value \times len(df) \times current\_churn\_rate \times potential\_churn\_reduction$$

Assuming a 10% reduction in churn, Big Retail could save approximately **$500,000 annually**. This financial metric provides a concrete and measurable target for the company's retention strategy, making a compelling case for investing in churn prevention initiatives. By demonstrating the direct financial benefits of reducing churn, Big Retail can allocate resources more effectively to initiatives that drive customer loyalty and long-term profitability.

```
for name, best_model, feature_importance_, results in training_results:
# Identify top churning customer segments
    print(f"\n***{name} Model:")
    top_churn_features = feature_importance_.head(3)['Feature'].tolist()
    churn_segments = df.groupby(top_churn_features)['Churn'].mean().sort_values(ascending=False).head(10)

    print("Top Churning Customer Segments:")
    print(churn_segments)

    # Calculate potential revenue saved by reducing churn
    avg_customer_value = df['TotalCharges_Original'].mean()
    print(f"\nAverage Customer Value: ${avg_customer_value:.2f}")
    current_churn_rate = df['Churn_Binary'].mean()
    print(f"Current Churn Rate: {current_churn_rate:.2f}")
    potential_churn_reduction = 0.1 # Assume we can reduce churn by 10%

    potential_savings = avg_customer_value * len(df) * current_churn_rate * potential_churn_reduction
    print(f"\nPotential Annual Revenue Saved: ${potential_savings:.2f}")
```

```
Top Churning Customer Segments:
Contract  TechSupport  tenure     PaperlessBilling  MultipleLines
0         2            -0.015113  1                 0              1.0
                                                    2              1.0
                       -0.910961  1                 2              1.0
                       -0.870241  0                 2              1.0
                       -0.788800  1                 1              1.0
                       -0.707359  1                 1              1.0
                       -0.666639  0                 0              1.0
                                  1                 1              1.0
                       -0.625919  0                 0              1.0
          0            -0.748080  0                 1              1.0
Name: Churn, dtype: float64

Average Customer Value: $2283.30
Current Churn Rate: 0.27

Potential Annual Revenue Saved: $426748.85
```

Business Insights and Recommendations

Based on the model's insights, the following business insights and actionable recommendations can be considered to reduce churn and improve customer retention:

Business Insights

- **Contract Type:** Short-term, flexible contracts are associated with higher churn, while longer-term contracts provide stability and reduce churn risk
- **Tech Support**: Lack of technical support is a major factor driving customers to leave. Enhancing or better promoting these services could help retain customers
- **Tenure:** New customers are the most vulnerable group for churn, indicating a need for targeted onboarding and engagement strategies

Recommendation

- **Promote Longer Contracts:** Offering discounts or incentives for customers to switch from month-to-month to annual contracts could reduce churn rates, as longer contracts are associated with increased customer retention.
- **Retention Programs for New Customers:** Focus on engaging customers within their first few months. Loyalty programs and personalized onboarding can help build long-term relationships.
- **Enhance Technical Support Services:** Big Retail should improve the quality and accessibility of its tech support services. This could include 24/7 support options, a more user-friendly help center, or proactive support offerings.
- **Focus on new customer engagement:** Implement a comprehensive onboarding and retention strategy for new customers. This could include personalized welcome offers, frequent engagement through targeted emails, and check-ins to ensure satisfaction within the first few months.
- **Targeted Campaigns for High-Risk Customers:** Use the model's predictions to develop campaigns targeting high-risk customers, offering them personalized deals to encourage retention.
- **Monitor and Optimize Retention Efforts**: Continuously monitor the effectiveness of these retention strategies and adjust based on customer feedback and changing trends

Conclusion

The predictive modelling project for **Big Retail** successfully identified key drivers of customer churn and provided actionable insights. By leveraging the **Decision Tree** and **Random Forest** models, the analysis can identify factors such as contract type, tenure, and tech support that significantly influence churn. The recommendations based on these findings offer clear strategies to reduce churn and improve customer retention. Implementing these strategies can lead to substantial cost savings and increased profitability. In the future, Big Retail can explore real-time analytics to predict churn as customers interact with the service. Integrating additional data sources, such as customer service interaction logs, could also enhance model performance and provide deeper insights into customer behaviour.

References

Carrington, A. M., Manuel, D. G., Fieguth, P. W., Ramsay, T., Osmani, V., Wernly, B., Bennett, C., Hawken, S., Magwood, O., Sheikh, Y., McInnes, M., & Holzinger, A. (2023). Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(1), 329–341. https://doi.org/10.1109/TPAMI.2022.3145392

De Caigny, A., Coussement, K., & De Bock, K. W. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. European Journal of Operational Research, 269(2), 760-772. https://doi.org/10.1016/j.ejor.2018.02.009

Ehsani, F., & Hosseini, M. (2024). Customer churn analysis using feature optimization methods and tree-based classifiers. Journal of Services Marketing, ahead-of-print(ahead-of-print). https://doi.org/10.1108/JSM-04-2024-0156

Khan, M. A. (2024). A Comparative Study on Imputation Techniques: Introducing a Transformer Model for Robust and Efficient Handling of Missing EEG Amplitude Data. Bioengineering, 11(8), 740.

Hidayat, A. (2024). Predictive Modelling of Liver Disease Using Biochemical Markers and K-Nearest Neighbors Algorithm. International Journal of Artificial Intelligence in Medical Issues, 2(2), 104-114.

Kim, S., & Lee, H. (2022). Customer churn prediction in influencer commerce: An application of decision trees. Procedia Computer Science, 199, 1332-1339. https://doi.org/10.1016/j.procs.2022.01.169

Liu, Y., Li, B., Yang, S., & Li, Z. (2024). Handling missing values and imbalanced classes in machine learning to predict consumer preference: Demonstrations and comparisons to prominent methods. Expert Systems with Applications, 237, 121694.

Rasheed, S., Kumar, G. K., Rani, D. M., Kantipudi, M. P., & Anila, M. (2024). Heart Disease Prediction Using GridSearchCV and Random Forest. EAI Endorsed Transactions on Pervasive Health and Technology, 10.

Rashmi, C. R., & Shantala, C. P. (2024). Evaluating Deep Learning with different feature scaling techniques for EEG-based Music Entrainment Brain Computer Interface. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 7, 100448.

Tang, J., Chen, W., Wang, K., Zhang, Y., & Liang, D. (2024). Probability-based label enhancement for multi-dimensional classification. *Information Sciences*, *653*, 119790.

Tiwaskar, S., Rashid, M., & Gokhale, P. (2024). Impact of machine learning-based imputation techniques on medical datasets-a comparative analysis. Multimedia Tools and Applications, 1-21.

Udandarao, V., & Gupta, P. (2024). Movie Revenue Prediction using Machine Learning Models. arXiv preprint arXiv:2405.11651.

Vujovic, Z. (2021). Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12, 599-606. https://doi.org/10.14569/IJACSA.2021.0120670

Zhu, W., Qiu, R., & Fu, Y. (2024). Comparative Study on the Performance of Categorical Variable Encoders in Classification and Regression Tasks. arXiv preprint arXiv:2401.09682.