BANA 273: MACHNE LRNG ANLYTCS

Group Project Report

Analyzing and Predicting Customer Churn for Telco Systems:

Insights and Strategies for Customer Retention

Team: 5A

Members: Andrew Sager, Rutul Bokade, Anthony Chang, Shyami Govind, Cynthia Wen

Contents

Executive Summary

Customer churn is a critical business problem that impacts industries worldwide. Retaining customers is often more cost-effective than acquiring new ones, making churn prediction a vital area of focus. Using the Telco Customer Churn dataset from Kaggle, we developed predictive models to identify the key factors influencing churn and determine actionable insights to help businesses reduce their churn rates.

Through the application of Random Forest, Logistic Regression, and XGBoost models, we evaluated the predictive accuracy and the impact of various preprocessing techniques, including feature scaling, resampling, and feature selection. Logistic Regression achieved the highest accuracy without preprocessing, while Random Forest and XGBoost provided valuable insights into feature importance and performance under different data treatments.

Our findings underscore the importance of Tenure, Monthly Charges, and Total Charges in predicting churn and suggest targeted strategies to retain customers.

Introduction

The business idea we have chosen to tackle is that of the churn rate of customers given a series of demographic and behavioral data. This is a business problem that pervades even the most simplistic of businesses and industries. Those looking to keep themselves in business almost always revert to the question: How can I gain customers and how can I maintain those that I already have? The latter is a question fundamental to business analytics in practice and is one that we will attempt to answer in the following report. To aid us in answering this question, we have collected data from the “Telco Customer Churn” dataset provided by Kaggle Community. We hope that this data, in conjunction with a series of predictive, machine learning, and subsequent ensemble models, will work together so that we can get a clearer picture of how we can predict the likelihood and outcome of whether a customer churns from a given business.

Data Summary and Visualization

Data Description

The dataset has 7043 records and includes information about the following:

- Churn – Customers who left the company withing the last month.
- Services that each customer utilizes, such as: phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.

- Account information for the customer – Tenure, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic information such as: gender, senior citizen or not, and if they have partners and dependents.

The data file itself is in the form of a .csv which was easily converted to Excel to be worked with. Came pre-cleaned, we did not necessarily have to go too in depth with the data scraping or cleaning, however we did have to transform the categorical variables when we were pre-processing the data in both the basic (scaling) and advanced (binning) fashions. The dataset had the following 21 columns:

Column	Description
customerID	Customer ID
gender	Whether the customer is a male or a female
SeniorCitizen	Whether the customer is a senior citizen or not (1, 0)
Partner	Whether the customer has a partner or not (Yes, No)
Dependents	Whether the customer has dependents or not (Yes, No)
tenure	Number of months the customer has stayed with the company
PhoneService	Whether the customer has a phone service or not (Yes, No)
MultipleLines	Whether the customer has multiple lines or not (Yes, No, No phone service)
InternetService	Customer's internet service provider (DSL, Fiber optic, No)
OnlineSecurity	Whether the customer has online security or not (Yes, No, No internet service)
OnlineBackup	Whether the customer has online backup or not (Yes, No, No internet service)
DeviceProtection	Whether the customer has device protection or not (Yes, No, No internet service)
TechSupport	Whether the customer has tech support or not (Yes, No, No internet service)
StreamingTV	Whether the customer has streaming TV or not (Yes, No, No internet service)
StreamingMovies	Whether the customer has streaming movies or not (Yes, No, No internet service)
Contract	The contract term of the customer (Month-to-month, One year, Two year)
PaperlessBilling	Whether the customer has paperless billing or not (Yes, No)
PaymentMethod	The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
MonthlyCharges	The amount charged to the customer monthly
TotalCharges	The total amount charged to the customer
Churn	Whether the customer churned or not (Yes or No)

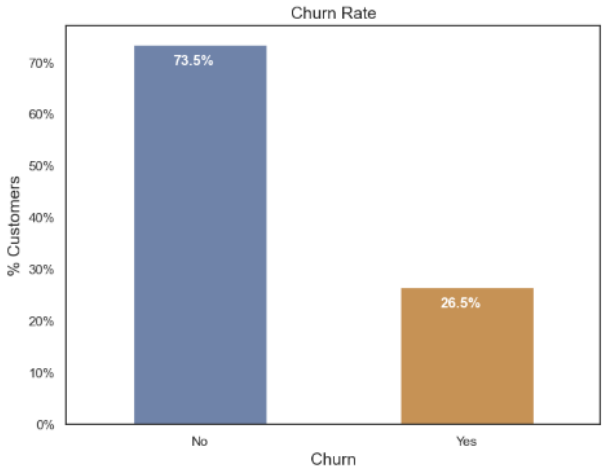
As previously stated, the Churn column is the attribute that we will be using for our dependent, or predicted, variable. Nearly all the attributes are categorical in nature, with the exception of Tenure, Monthly Charges, and Total Charges.

In analysis, after ensuring there were no repeats, we excluded the Customer ID variable, as we decided that knowing this would not be helpful to us for deciding factors that relate to churn rate and in fact could have led to overfitting had we kept it. For some reason, the Senior Citizen

categorical variable did not indicate status with a “Yes” or “No”, but instead took a binary “0, 1” approach, which contrasted with the rest of the binary variables. Subsequently, we ensured that the variables were uniform in this sense before we did any further analysis.

Exploratory Data Analysis: Visualization

The visualizations done to represent this data were done exclusively via Python and focused primarily on the interactions each of the attributes had on the target variable of churn. The intention of these visualizations was to explore the dataset and understand patterns and relationships between various attributes that might be influencing the target outcome (Churn).



Churn Distribution: In the data, 73.5% of the customers do not churn. Thus, the data is skewed and it’s important to take this into consideration while modelling as this skewedness could lead to a lot of false negatives. This imbalance highlights the need for resampling techniques like SMOTE to address minority class bias.

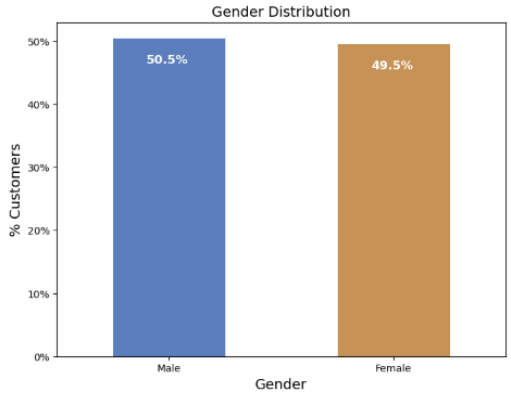
Churn by Different Demographic Attributes:

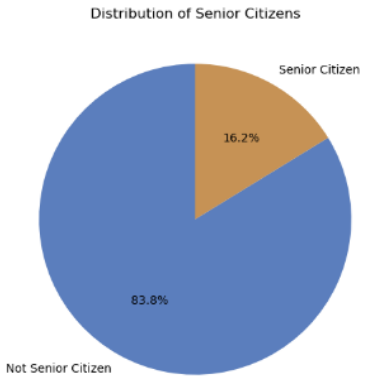
Churn across gender types: The distribution of the churn variable is balanced across the gender types, with 50.5% males and 49.5% females. Thus, it seems that gender doesn’t have a high impact on the churn rates.

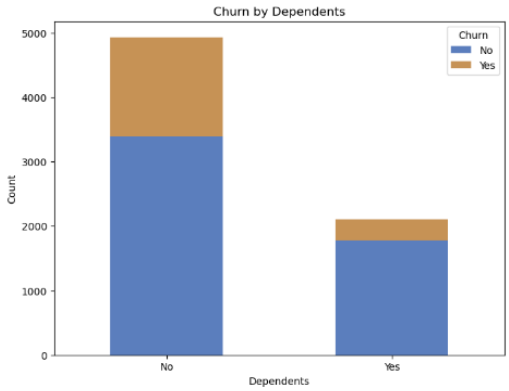
Churn across Senior citizen: 83.8% of the customers are not senior citizens, thus senior citizens form a minority of the customer base and might need targeted analysis.

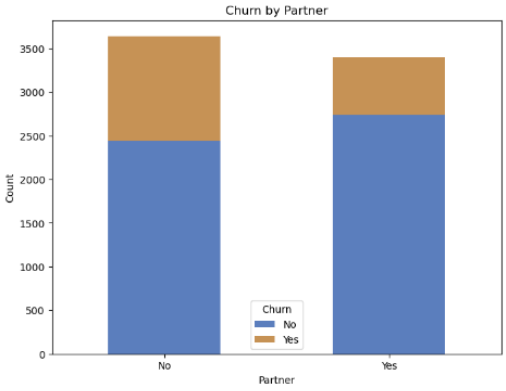
Churn by Dependents: The plot indicates that customers without dependents churn at a higher rate compared to those with dependents. This insight could be useful for forming effective strategies for effective targeting of customers without dependents.

Churn by Partner: There seems to be only a slight difference between customers with and without partners.



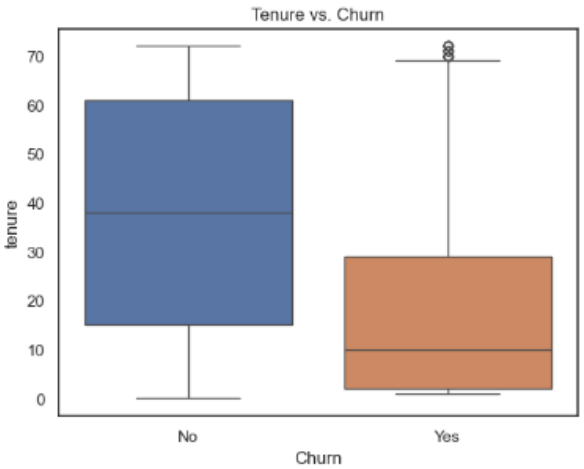






Churn by Different Account Information Attributes:

Tenure vs. Churn: The box plot of churn against the tenure helped identify how length of time a customer has been with the company is related to the churn behavior. It revealed that customers with shorter tenure with the telecom company are more likely to churn.

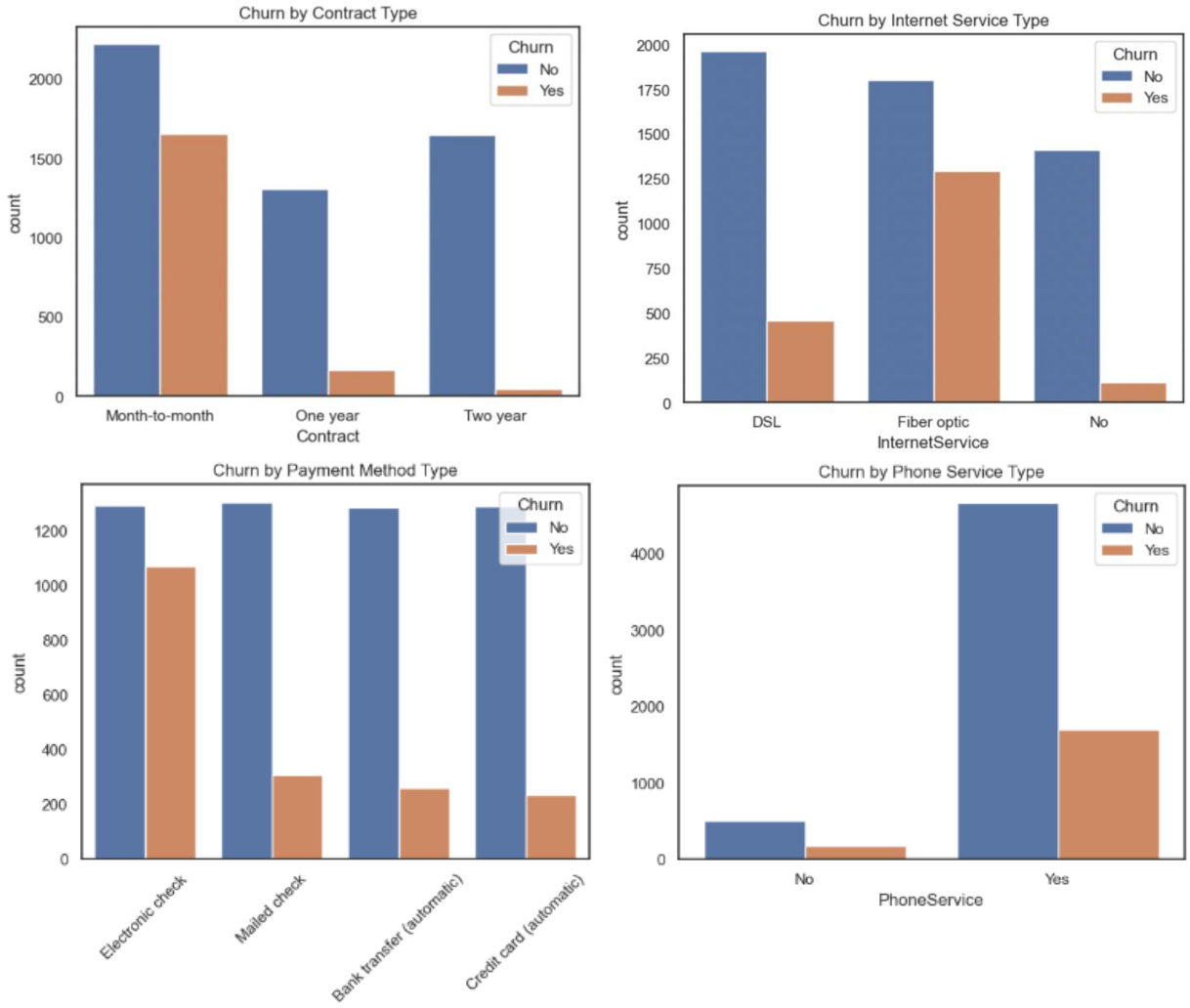


Churn rates across Contract Types: Helped identify how type of customer contract influences the churn rate. Count plots showed that customers with month-to-month contracts churn more often than those with longer-term contracts.

Churn rates across Internet service Types: Explores whether the type of internet service (e.g., DSL, Fiber Optic, No Internet) impacts churn. The graph indicated that customers using Fiber Optic services exhibit a higher churn rate compared to DSL users.

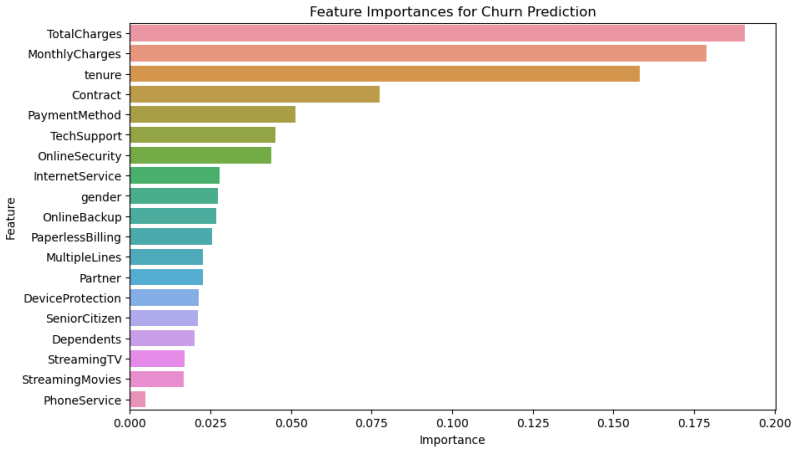
Churn rates across Payment Method Types: Helped identify how the method of payment could indicate the probability of churn. The plots indicated that customers making a payment via an Electronic Check are most likely to churn compared to customers who use other methods of payment.

Churn rates across Phone Service Types: Indicates the relation between phone service type and the likelihood to churn. The plot indicated that customers with Phone service are most likely to 'Not Churn'.



Feature importance in prediction:

We found that the attributes of Total Charges, Monthly Charges, and tenure all can significantly indicate whether a customer is likely to churn.

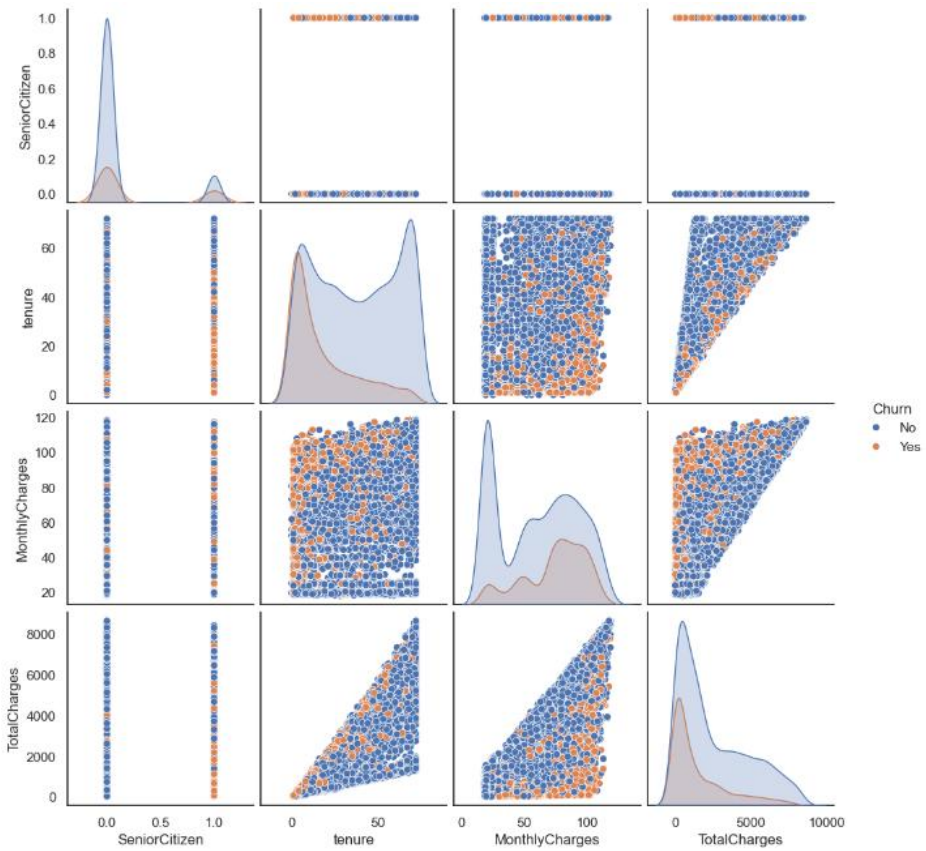


Pairwise Relationships:

We used a pairplot to visualize pairwise relationship between the numerical features in the dataset, with the points color coded by the target 'Churn' variable.

The key findings from this plot are:

- Most customers are not senior citizens.
- Tenure emerges as an important feature; long-tenured customers appear to be loyal.
- Customers with high monthly charges tend to be churning more.
- Long term customers with high total charges are less likely to churn.

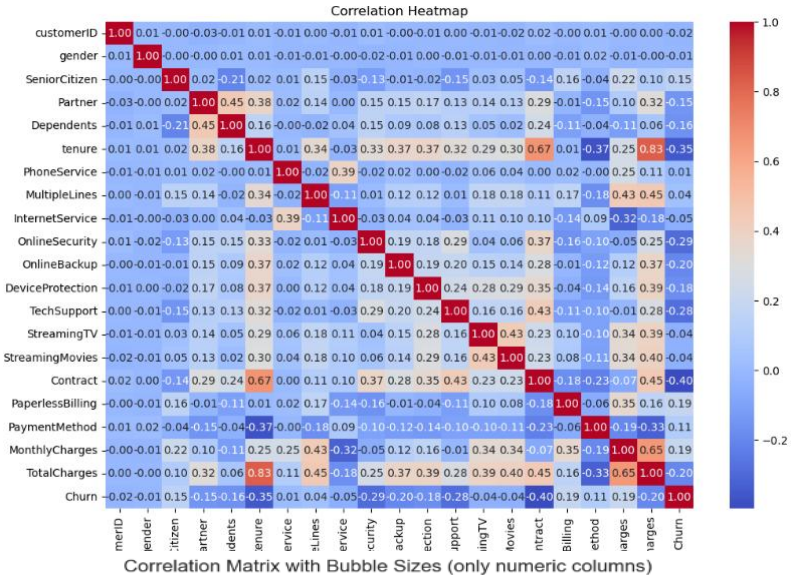


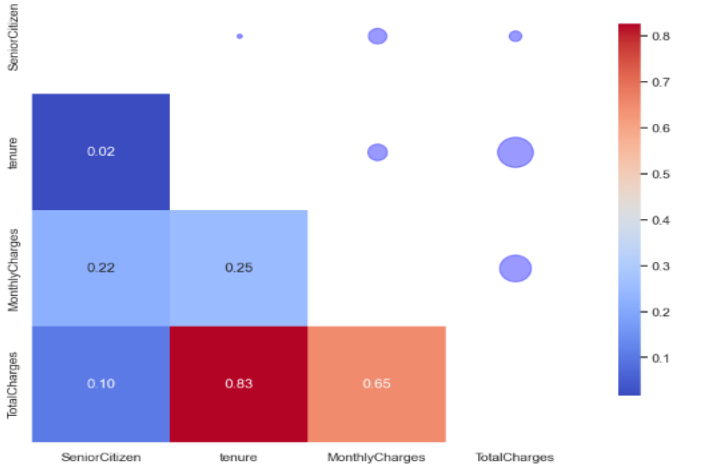
Feature Correlation:

We analyzed the correlation heatmap of each of the features to determine if we would have a collinearity issue while attempting to run the logit model on our data. We found that:

- Tenure was slightly correlated with Contract and Total Charges naturally, as we would generally expect them to increase along with each other.
- Similarly, Monthly Charges and Total Charges were correlated, but expected to increase in the same manner.

Outside of these few correlations, though, we were pleased to find no other relationships with a coefficient above 0.5, indicating strong promise for the efficacy of the logit model in our eventual analysis.





Analysis

Process

Following are the main data limitations/features we would need to consider while evaluating our models:

1. We have a very skewed Target class proportion, with an unbalanced distribution of approximately 75%-25%.
2. Some features would intuitively be correlated, and the model or the pre-processing steps should be able to handle that.

3. For our business case, where the goal is to prevent customer loss, we should aim to ensure that most churners get identified, even at the cost of misclassifying a few non-churners. This is because missing out on flagging each churner could translate into loss in revenue and a loyal customer base.

For the above-mentioned reasons, we recognized that accuracy alone can't be a good metric for evaluating the performance of our models. We decided to focus on **Recall** (with 1/'Yes" as the positive class), **Accuracy** and **Expected Value** (cost matrix details in Appendix 1) to get a comprehensive view of the model's performance.

The dataset involves a binary classification problem, i.e. predicting churn or no churn, and has a mix of categorical and numerical attributes. We choose to build and compare the following 3 models for their below mentioned strengths. The models were trained with a split of 80-20 for the train and the test subsets.

Logistic Regression

- It's easily understandable being a linear model, that works well with binary classification problems.
- It's computationally efficient.
- It works as a great baseline for comparison with more complex models.

Limitations: would require some preprocessing like encoding and scaling to hand the numerical and categorical variables.

XGBoost

- It's good at capturing complex and non-linear relationships between features.
- Helps identify the most important features influencing the churn.
- It can handle missing values without requiring any pre-processing.
- Efficient at handling large datasets.
- Effectively handles the class imbalance in the target class.

Limitations: Computationally expensive and requires careful tuning.

Random Forest

- Can capture non-linear relationships.
- Works with both categorical and numerical variables.
- Handles correlated variables effectively.
- Similar to XGBoost, it highlights feature importances indicating which features are most important for predicting churn.

Limitations: The model is harder to interpret.

Benchmarks

Without Preprocessing:

- Logistic Regression achieved 82.39% **accuracy,** a **recall** of 60% and a negative **expected value** of -$60750, outperforming other models in initial testing.
- Churn class proportions:
  - Non-churn (No): 73.46%
  - Churn (Yes): 26.54%.

Baseline Observations:

- All models demonstrated a bias toward the majority class due to the imbalanced dataset.

Preprocessing and Iterative Analysis

We built the models initially with minimal preprocessing and used 10 fold stratified cross validation to evaluate the accuracy and recall of the models. Then we went on to perform incremental pre-processing steps and observing how that impacted the cross validated accuracy and recall for each of the model and then finally used a combination of the discussed preprocessing steps to understand its impact on the performance of the model. The tables below give a detailed description of our findings.

Preprocessing Steps	Recall		
	Logistic Regression	XGBoost	Random Forest
without pre-processing (only encoding categorical values)	0.6	0.53	0.44
Hyperparameter Tuning	NA	NA	0.43
Minimal Preprocessing (+missing values)	0.6	0.54	0.7353
Individual Preprocessing			
Standard Scaler	0.6	0.54	0.5
MinMax Scaler	0.6	0.54	0.5
Evaluating with Feature Selection			
SelectKBest with top 10 features	0.51	0.56	0.57
Binning Continuous Features	0.59	0.51	0.5
Evaluating with Different Resampling Methods			
SMOTE	0.78	0.51	0.5
RandomOverSampler	0.82	0.51	0.5
Combined Preprocessing			
Combined Pre-processing (All Steps)	0.8	0.85	0.84
Combined Pre-Processing (One-Hot Encoding + missing values with median +feature scaling with Standard scaler +SMOTE)	0.83	0.68	0.72

Preporocessing Steps	Expected Value		
	Logistic Regression	XGBoost	Random Forest
without pre-processing (only encoding categorical values)	-43.12	NA	NA
Individual Preprocessing			
Standard Scaler	-4.93	-20.83	-28.28
MinMax Scaler	-4.83	-20.83	-28.28
Evaluating with Feature Selection (SelectKBest)			
SelectKBest with top 10 features	-27.01	-14.48	-11.92
Binning Continuous Features	-6.28	-28.28	-28.46
Evaluating with Different Resampling Methods			
SMOTE	39.18	-28.28	-28.46
RandomOverSampler	49.36	-28.28	-28.46
Combined Preprocessing			
Combined Pre-processing (All Steps)	96.16	117.46	112.51

Preprocessing Steps	Accuracy Scores					
	Logistic Regression	10 Fold Cross Validated (mean Accuracy)	XGBoost	10 Fold Cross Validated (mean Accuracy)	Random Forest	10 Fold Cross Validated (mean Accuracy)
Without pre-processing (only encoding categorical values)	0.8239	0.8021	0.8076	0.7979	0.7977	0.7968
Hyperparameter Tuning	NA	NA	NA	NA	0.7984	0.7985
Minimal Preprocessing (+missing values)	0.824	0.802	0.8077	0.7979	0.7353	0.7346
Individual Preprocessing						
Standard Scaler	0.8197	0.7994	0.7984	0.7729	0.8112	0.8015
MinMax Scaler	0.8219	0.8001	0.7984	0.7729	0.8112	0.8019
Evaluating with Feature Selection						
SelectKBest with top 10 features	0.8098	0.7903	0.8041	0.7795	0.8013	0.7761
Binning Continuous Features	0.8197	0.8019	0.7842	0.7721	0.8077	0.7987
Evaluating with Different Resampling Methods						
SMOTE	0.7715	0.7964	0.7842	0.7721	0.8077	0.7987
RandomOverSampler	0.7594	0.762	0.7842	0.7721	0.8077	0.7987
Combined Preprocessing						
Combined Pre-processing (All Steps)	0.7691	0.7531	0.7821	0.7658	0.7841	0.7725
Combined Pre-Processing (One-Hot Encoding + missing values with median +feature scaling with Standard scaler +SMOTE)	0.7551	0.7499	0.7999	0.7919	0.7892	0.7816

Key Steps and Observations:

- *Without Pre-Processing*
  - Logistic regression shows the highest cross-validated accuracy of 80.21% and a recall of 60% with just basic encoding. XGBoost and Random Forest are also relatively robust, but with lower recall without preprocessing.
  - Processing the missing values very minimally improves the accuracy for Logistic regression but doesn't show much improvement for the other two models. As for the Recall, it significantly improved in the Random Forest model.
- *Scaling*:
  - MinMaxScaler slightly improved Logistic Regression accuracy to 82.19%, with consistent performance across models. The recall didn't show a lot of improvement.
  - It is important to scale the variables in logistic regression so that all of them are within a range of 0 to 1. Scaling ensured numerical features like Monthly Charges and Total Charges were on a comparable scale, improving model stability.

- *Resampling (SMOTE and RandomOverSampler):*
  - SMOTE improved churn recall to 78%, but overall accuracy dropped to 79.64% for Logistic Regression.
  - RandomOverSampler yielded similar recall improvements but slightly lower accuracy (76.2%).

- *Feature Selection (SelectKBest):*
  - Reducing features to the top 10 using SelectKBest reduced accuracy to 80.98%, highlighting the importance of retaining key features like Tenure, Monthly Charges, and Contract Type. But the Recall stayed low.

- *Binning Continuous Features:*
  - Binning Tenure and Monthly Charges into categories improved interpretability and maintained accuracy at 81.97%, but still the Recall remained low.

- *Combined Preprocessing:*
  - Combining MinMaxScaler, SelectKBest, and SMOTE produced 76.91% **accuracy**, showing that overly complex preprocessing pipelines may not always yield the best results in accuracy, but improved the Recall

Model Evaluation

The table summarizes performance metrics of the best performing version of each model (details in Appendix 2). The findings from each are discussed below:

Model	Accuracy (Cross validated)	Precision (Churn)	Recall (Churn)	F1-Score (Churn)	Expected Value
Logistic Regression (Combined Pre-processing)	75.31%	76%	80%	78%	$96.16
Random Forest (Combined Pre-processing)	77.25%	76%	84%	80%	$117.46
XGBoost (Combined Pre-processing)	76.58%	75%	85%	80%	$112.51

1. **Logistic Regression (with SMOTE)**
   - Delivers the highest accuracy of 79.64%, but this might not be an accurate measure of performance on the generalization dataset, since it could be misleading in imbalanced datasets.
   - Has a decent expected value, but lower compared to the other two, which could be attributed to higher false negatives respect to the other two models.
   - The Recall is decent, thus minimizing the false negatives (failing to identify a churner).
   - ***This model would work fine, if the cost of targeting customers falsely identified as churners is not very high and is not a business concern.***

2. **Random Forest (Combined Pre-processing)**
   - This model gives a very balanced performance with decent accuracy (77.25%) combined with a high precision (76%) and recall (84%), thus resulting in an improved F1-score (80%).
   - Delivers highest expected value due to lowest false negatives out of the three top models.
   - This indicates that this is a robust model, making confident predictions, with fewer false positives and good recall and expected value indicative of lower false negatives.
   - ***This model is great when the cost of missing a churner is high, i.e. works best in scenarios where the requirement is to correctly identify churners and target them.***

3. **XGBoost (Combined Pre-processing)**
   - Very similar to the Random Forest, with high precision (75%), recall (85%) leading to a strong F1 score (80%) and good expected value.

Best Model: Random Forest with combined Pre-processing due to its balanced performance and suitability for the Telco Churn business case.

Recommendations

1. *Customer Retention Strategies:*
   a. **Prioritize Recall and Expected Value**, even if it might come with a minor trade off in accuracy or precision, since missing out on the churners (false negatives) will be costlier to the business than targeting loyal customers.
   b. **Focus on short-tenure customers** by offering incentives or personalized plans to increase loyalty.
   c. **Target customers with high Monthly Charges** by reviewing pricing strategies or offering value-added services.

2. *Future Model Improvements:*
   a. **Tuning the Models** - Experiment with hyperparameter tuning for Random Forest and XGBoost to improve recall.
   b. **Explore advanced feature engineering techniques**, such as grid search, interaction terms or clustering.
   c. **Explore other Resampling techniques**

3. *Business Implementation:*
   a. **Churn monitoring Dashboards**: Develop dashboards to monitor key churn indicators like Tenure and Monthly Charges in real time. This will help the stakeholders make timely decisions.
   b. **Use churn predictions** to prioritize outreach to at-risk customers.

These recommendations are targeted at balancing model improvements and its real-world implementation. The insights gained from the exploratory data analysis, coupled with the model building and evaluation process, have been used coherently to devise actionable strategies. These strategies could leverage the dataset at hand to create effective business plans that could help in predicting Churn, establish mechanisms to monitor and take actions to manage and reduce the Churn rate effectively.

Conclusion-Key learnings from Data Mining

This analysis demonstrates the power of machine learning in addressing the critical business problem of customer churn. The whole process provided valuable insights on how understanding the data is an important precursor to developing a strong model. It also highlighted how the model evaluation needs to be varied based on the business case at hand and always Accuracy could not be considered the primary indicator of a model performance and robustness.

Key Takeaways:

- Logistic Regression initially emerged as the best model due to its simplicity and high accuracy.
- Preprocessing techniques highlighted the importance of balancing recall, accuracy and expected value .
- By leveraging these insights, businesses can make data-driven decisions to retain customers and improve profitability.
- The analysis indicates how machine learning tools explored throughout can effectively turn raw data into actionable business insights that can support strategic decision making.
- Real time monitoring of key attributes and constant model refinement can enhance the performance of businesses by pushing up profits, fostering customer loyalty and gaining a competitive advantage in the market.

Appendix 1

```
# Perform Expected value Calculations
y_true = y_test
# Define cost/profit values
profit_true_positive = 400   # Profit for correctly identified churners
cost_false_positive = -50    # Cost for falsely identified churners
cost_false_negative = -600   # Cost for missed churners

# Function to calculate Expected value from a confusion matrix
tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()
cm_sum = tn + fp + fn + tp
expected_value = ((tp/cm_sum) * profit_true_positive + (fp/cm_sum) * cost_false_positive + (fn/cm_sum) * cost_false_negative)
print(f"Expected value: {expected_value:.4f}")
```

Appendix 2

Logistic Regression (Combined Pre-processing)

```
Using Combined Pre-processing: MinMaxScaler + SelectKBest + SMOTE
Expected value using LogisticRegression(max_iter=1000, random_state=42): 96.1594
Accuracy using LogisticRegression(max_iter=1000, random_state=42): 0.7691
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.73      0.76      1021
           1       0.76      0.80      0.78      1049

    accuracy                           0.77      2070
   macro avg       0.77      0.77      0.77      2070
weighted avg       0.77      0.77      0.77      2070

Confusion Matrix:
 [[750 271]
 [207 842]]
Cross-Validation Scores LogisticRegression(max_iter=1000, random_state=42): [0.73309179 0.75362319 0.78502415 0.75966184 0.74396135 0.73913043
 0.73792271 0.79468599 0.74244256 0.74244256]
Mean CV Accuracy: 0.7531986576240295
Standard Deviation of CV Accuracy: 0.01981990085143312
```

Random Forest (Combined Pre-processing)

```
Expected value using RandomForestClassifier(max_features='log2', min_samples_leaf=4,
                       n_estimators=200, random_state=42): 112.5121
Accuracy using RandomForestClassifier(max_features='log2', min_samples_leaf=4,
                       n_estimators=200, random_state=42): 0.7841
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.73      0.77      1021
           1       0.76      0.84      0.80      1049

    accuracy                           0.78      2070
   macro avg       0.79      0.78      0.78      2070
weighted avg       0.79      0.78      0.78      2070

Confusion Matrix:
 [[747 274]
 [173 876]]
Cross-Validation Scores RandomForestClassifier(max_features='log2', min_samples_leaf=4,
                       n_estimators=200, random_state=42): [0.76570048 0.76086957 0.78985507 0.77898551 0.75120773 0.77536232
 0.75241546 0.79710145 0.77629988 0.77750907]
Mean CV Accuracy: 0.7725306532545899
Standard Deviation of CV Accuracy: 0.014263432244224718
```

XGBoost (Combined Pre-processing)

```
Expected value using XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='logloss',
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None,
              n_jobs=None, num_parallel_tree=None, random_state=42, ...): 117.4638
Accuracy using XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='logloss',
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None,
              n_jobs=None, num_parallel_tree=None, random_state=42, ...): 0.7821
Classification Report:
               precision    recall  f1-score   support

           0       0.82      0.72      0.76      1021
           1       0.75      0.85      0.80      1049

    accuracy                           0.78      2070
   macro avg       0.79      0.78      0.78      2070
weighted avg       0.79      0.78      0.78      2070

Confusion Matrix:
 [[732 289]
 [162 887]]


Cross-Validation Scores XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='logloss',
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None,
              n_jobs=None, num_parallel_tree=None, random_state=42, ...): [0.76328502 0.75845411 0.78623188 0.76570048 0.75603865 0.7705314
 0.75241546 0.78502415 0.75816203 0.76299879]
Mean CV Accuracy: 0.7658841981669384
Standard Deviation of CV Accuracy: 0.011001171351752116
```