Advanced Telco Customer Churn Prediction

1. Project Overview and Objectives

This project focuses on developing a robust and accurate machine learning solution to predict customer churn in a telecommunications company. The primary objective is to identify customers at high risk of churning, enabling proactive intervention strategies to improve customer retention and maximize Customer Lifetime Value (CLV). By leveraging advanced data processing, feature engineering, and ensemble modeling techniques, this project aims to provide actionable insights that drive significant business impact.

Key objectives include:

- **Accurate Churn Prediction:** Develop a highly accurate predictive model to identify potential churners.
- **Actionable Insights:** Provide insights into the key drivers of churn to inform retention strategies.
- **Business Impact Quantification:** Quantify the financial benefits of churn reduction, demonstrating a clear Return on Investment (ROI).
- **Scalable Solution:** Create a modular and reproducible pipeline for data processing and model deployment.
- **Comprehensive Documentation:** Offer thorough documentation for understanding, using, and extending the project.

2. Dataset Information and Key Features

The project utilizes a comprehensive dataset containing various customer attributes and service usage patterns from a telecommunications company. The dataset is typically structured with each row representing a unique customer and columns detailing their demographic information, services subscribed, monthly charges, total charges, and churn status.

Dataset Source: `WA_Fn-UseC_-Telco-Customer-Churn.csv` located in `data/raw/`.

Key Features (examples, actual features may vary slightly based on data exploration):

- **Demographic Information:** `gender`, `SeniorCitizen`, `Partner`, `Dependents`
- **Account Information:** `tenure`, `Contract`, `PaperlessBilling`, `PaymentMethod`, `MonthlyCharges`, `TotalCharges`
- **Services Subscribed:** `PhoneService`, `MultipleLines`, `InternetService`, `OnlineSecurity`, `OnlineBackup`, `DeviceProtection`, `TechSupport`, `StreamingTV`, `StreamingMovies`
- **Target Variable:** `Churn` (Yes/No, indicating whether the customer churned)

This dataset undergoes extensive preprocessing, including handling missing values, outlier detection, feature encoding, and scaling, to prepare it for model training.

3. Project Structure Explanation

The project is organized into a well-defined directory structure to ensure modularity, maintainability, and ease of navigation. Below is an overview of the main directories and their contents:

```
. # Project Root
   artifacts/             # Stores processed data, trained models, and various reports
      data/               # Cleaned and split datasets (X_train, X_test, y_train, y_test)
      encode/             # Encoded artifacts (e.g., label encoders)
      models/             # Trained machine learning models (e.g., Random Forest, XGBoost)
      scalers/            # Scaler objects (e.g., StandardScaler)
   config/                # Configuration files for the project
      config.yaml         # Centralized configuration for paths, parameters, etc.
   data/                  # Raw and processed data files
      processed/          # Intermediate processed data files
      raw/                # Original raw dataset
   logs/                  # Log files generated during pipeline execution
   notebooks/             # Jupyter notebooks for exploratory analysis, model development, a
      0_data_preparation.ipynb
      1_exploratory_data_analysis.ipynb
      2_feature_engineering.ipynb
      3_model_development.ipynb
      4_ensemble_methods.ipynb
      5_model_evaluation.ipynb
      6_business_analysis.ipynb
      artifacts/          # Notebook-specific artifacts (data, models, processed files, repor
   pipelines/             # Python scripts for the automated data processing and model train
      data_pipeline.py    # Main script for running the end-to-end pipeline
   src/                   # Source code for data processing, utilities, and modeling
      data_processing/    # Modules for data ingestion, cleaning, feature engineering
      utils/              # Utility functions (e.g., logging, configuration management)
   requirements.txt       # List of Python dependencies
```

4. Installation and Usage

Installation

To set up the project locally, follow these steps:

1. **Clone the repository:**

   ```
   git clone <repository_url>
   cd AdvanceTelcoCustomerChurnPrediction
   ```

(Note: Replace `<repository_url>` with the actual URL of the repository.)

2. **Create a virtual environment (recommended):**

   ```
   python3 -m venv venv
   source venv/bin/activate  # On Windows, use `venv\Scripts\activate`
   ```

3. **Install dependencies:** `bash     pip install -r requirements.txt`

Usage

Jupyter Notebooks The `notebooks/` directory contains a series of Jupyter notebooks that walk through the entire project lifecycle, from data preparation to business analysis. To run these notebooks:

1. **Start Jupyter Lab:**

   ```
   jupyter lab
   ```

2. Navigate to the `notebooks/` directory in the Jupyter Lab interface.
3. Open and run the notebooks sequentially:
   - `0_data_preparation.ipynb`: Handles initial data loading and cleaning.
   - `1_exploratory_data_analysis.ipynb`: Performs in-depth EDA to understand data patterns.
   - `2_feature_engineering.ipynb`: Develops new features from existing ones.
   - `3_model_development.ipynb`: Trains and evaluates various machine learning models.
   - `4_ensemble_methods.ipynb`: Explores and implements ensemble modeling techniques.
   - `5_model_evaluation.ipynb`: Provides comprehensive model evaluation metrics.
   - `6_business_analysis.ipynb`: Translates model results into business insights and ROI.

Python Data Processing Pipeline The `pipelines/data_pipeline.py` script provides an automated way to run the entire data processing and model training workflow. This script is designed for reproducibility and can be integrated into production environments.

To run the pipeline:

1. **Ensure all dependencies are installed** (as per the Installation section).
2. **Execute the pipeline script:** `bash     python pipelines/data_pipeline.py`

This script will:

- Load raw data.

- Perform data cleaning and preprocessing.
- Engineer new features.
- Train the best-performing model (Random Forest, as identified in the notebooks).
- Save trained models and relevant artifacts to the `artifacts/` directory.
- Generate logs in the `logs/` directory.

Note: The `config/config.yaml` file can be modified to adjust pipeline parameters, such as model hyperparameters or data paths.

5. Key Results and Business Impact

This project successfully developed a predictive model capable of identifying churning customers with high accuracy and significant business impact. The key findings and their implications are summarized below:

Model Performance

- **Best Performing Model:** Random Forest
- **F1 Score:** 0.619
- **ROC-AUC:** 0.845
- The model effectively identifies **74.9%** of churning customers, providing a strong basis for targeted retention efforts.

Business Impact Summary

By implementing the recommendations derived from this project, the telecommunications company can expect substantial financial benefits:

- **Current Annual Churn Loss:** Approximately **$291,720**.
- **Customers at Risk:** 374 customers, representing a 26.5% churn rate within the analyzed segment.
- **Customer Lifetime Value (CLV) at Risk:** A significant **$593,893** in potential lost revenue.

Recommended Strategy: Target All Predicted Churners

- **Target Audience:** 530 customers identified as high-risk churners.
- **Estimated Investment for Retention Campaigns:** $18,250.
- **Expected Customer Retention:** 84 customers.
- **Projected Revenue Impact (Saved):** $65,520.
- **Return on Investment (ROI): 259.0%**
- **Net Benefit:** A substantial **$47,270**.

This strategy demonstrates a highly favorable ROI with a payback period of less than 6 months, making it a financially sound decision for the business.

High-Priority Customer Segments

The analysis identified several customer segments that warrant particular attention for retention efforts:

1. **High-Value Loyal Customers:** These customers have the highest Customer Lifetime Value but present a moderate churn risk. Retaining them is crucial for long-term profitability.
2. **High-Value New Customers:** Despite being new, these customers exhibit premium pricing and the highest churn risk, indicating a need for early engagement and satisfaction monitoring.
3. **Fiber Optic Customers:** Customers subscribed to Fiber Optic services show elevated churn rates, suggesting potential issues related to service quality, pricing, or support for this specific offering.
4. **Month-to-Month Contracts:** Customers on flexible month-to-month contracts have the highest churn risk due to ease of switching, necessitating proactive engagement and value reinforcement.

Expected Outcomes (Year 1 Projections)

- **Churn Reduction:** Retention of 84 customers.
- **Revenue Protection:** $65,520 in revenue protected from churn.
- **Campaign Investment:** $18,250 allocated for retention efforts.
- **Net Positive Impact:** A significant $47,270 financial gain.
- **CLV Protection:** $131,040 in Customer Lifetime Value protected.

Risk Mitigation

To ensure the continued success and effectiveness of the churn prediction model and retention strategies, the following risk mitigation measures are recommended:

- **Model Monitoring and Retraining:** Implement a schedule for model monitoring and retraining every 3 months to account for changes in customer behavior and market dynamics.
- **A/B Testing Framework:** Establish an A/B testing framework for retention campaigns to continuously optimize their effectiveness and identify the most impactful strategies.
- **Gradual Rollout:** Implement a gradual rollout of retention initiatives to minimize operational risks and allow for adjustments based on early feedback.
- **Customer Satisfaction Monitoring:** Continuously monitor customer satisfaction metrics to identify emerging issues and proactively address potential churn triggers.

Recommendations

Based on the comprehensive analysis, the following recommendations are put forth:

1. **Immediate Action:** Deploy the Random Forest model for churn prediction without delay.
2. **Target Strategy:** Focus retention efforts on the segment of customers identified as

high-risk churners. 3. **Investment:** Allocate the recommended $18,250 for initial retention campaigns. 4. **Success Metrics:** Implement robust tracking for key metrics including churn rate, customer retention rate, ROI, and customer satisfaction to measure the ongoing impact of the churn prediction system. 5. **Long-term Strategy:** Expand the scope to include proactive customer value optimization, moving beyond just churn prevention to maximizing overall customer lifetime value.

Expected ROI: 259.0% with a payback period of less than 6 months.

6. Technical Approach

This project employs a robust technical approach encompassing advanced machine learning techniques to achieve high predictive accuracy and actionable insights. The core components include:

Data Preprocessing and Feature Engineering

- **Handling Missing Values:** Various strategies are applied to address missing data, ensuring data integrity.
- **Outlier Detection:** Techniques are used to identify and manage outliers, preventing skewed model performance.
- **Feature Encoding:** Categorical features are transformed into numerical representations suitable for machine learning algorithms.
- **Feature Scaling:** Numerical features are scaled to standardize their range, which is crucial for many algorithms.
- **Feature Binning:** Continuous numerical features are converted into discrete bins to capture non-linear relationships and reduce noise.

Model Selection and Training

- **Ensemble Methods:** The project extensively utilizes ensemble learning techniques, which combine multiple models to achieve better predictive performance than could be obtained from any of the constituent models alone. This includes:
  - **Random Forest:** A powerful ensemble method that builds multiple decision trees and merges their predictions to improve accuracy and control overfitting. This model was identified as the best performer.

  - **XGBoost:** An optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework.
  - **CatBoost:** A gradient boosting library developed by Yandex. It is known for its excellent performance with categorical features and robustness.
  - **Voting Ensemble:** A meta-estimator that trains a collection of diverse base estimators and then predicts by averaging the individual predictions. This approach helps to reduce variance and improve generalization.
- **Logistic Regression with SMOTE:** Logistic Regression is used as a baseline model, and SMOTE (Synthetic Minority Over-sampling Technique) is applied to address class imbalance, ensuring the model does not disproportionately favor the majority class.

Class Imbalance Handling

Customer churn datasets often exhibit class imbalance, where the number of non-churning customers significantly outweighs churning customers. To address this, the project incorporates techniques such as:

- **SMOTE (Synthetic Minority Over-sampling Technique):** Used to generate synthetic samples for the minority class (churners), thereby balancing the dataset and improving the model's ability to learn from churn patterns.
- **Adjusted Class Weights:** Models are trained with adjusted class weights to penalize misclassifications of the minority class more heavily.
- **Performance Metrics:** Evaluation focuses on metrics suitable for imbalanced datasets, such as F1-Score and ROC-AUC, rather than simple accuracy.

7. File Descriptions and Workflow

File Descriptions

- `WA_Fn-UseC_-Telco-Customer-Churn.csv`: The raw dataset containing customer information.
- `config/config.yaml`: Configuration file for defining paths, parameters, and other settings.
- `notebooks/`: Contains Jupyter notebooks for each stage of the project.
  - `0_data_preparation.ipynb`: Initial data loading, cleaning, and preprocessing steps.
  - `1_exploratory_data_analysis.ipynb`: Detailed analysis of data distributions, correlations, and insights.
  - `2_feature_engineering.ipynb`: Steps for creating new features and transforming existing ones.

  - `3_model_development.ipynb`: Training and evaluation of individual machine learning models.
  - `4_ensemble_methods.ipynb`: Implementation and comparison of ensemble models.
  - `5_model_evaluation.ipynb`: Comprehensive evaluation of the best models using various metrics.
  - `6_business_analysis.ipynb`: Translation of model results into business insights, ROI calculation, and strategic recommendations.
- `pipelines/data_pipeline.py`: The main Python script that orchestrates the entire data processing, model training, and evaluation workflow.
- `src/data_processing/`: Python modules for specific data processing tasks (e.g., `data_ingestion.py`, `handle_missing_values.py`, `feature_encoding.py`, `feature_scaling.py`, `feature_binning.py`, `outlier_detection.py`, `data_splitter.py`).
- `src/utils/`: Utility functions, including `config.py` for configuration management and `logger.py` for logging.
- `artifacts/`: Directory for saving processed data, trained models, encoders, scalers, and reports.
  - `artifacts/models/`: Stores the trained machine learning models (e.g., `random_forest_optimized.pkl`).
  - `artifacts/reports/executive_summary.md`: A summary of key findings and business impact.

Workflow

The typical workflow for this project involves:

1. **Data Ingestion:** Loading the raw `WA_Fn-UseC_-Telco-Customer-Churn.csv` dataset.
2. **Data Preprocessing:** Applying cleaning, handling missing values, and outlier detection.
3. **Feature Engineering:** Creating new features and transforming existing ones to enhance model performance.
4. **Data Splitting:** Dividing the dataset into training and testing sets.
5. **Model Training:** Training various machine learning models, including ensemble methods, on the preprocessed data.
6. **Model Evaluation:** Assessing model performance using appropriate metrics (F1-Score, ROC-AUC).
7. **Business Analysis:** Translating model predictions into actionable business strategies and quantifying ROI.
8. **Pipeline Execution:** Running the `data_pipeline.py` script to automate the end-to-end process.

8. Requirements and Dependencies

This project requires Python 3.8+ and the following libraries. All dependencies are listed in `requirements.txt` and can be installed using pip:

```
pip install -r requirements.txt
```

Key libraries include:

- `pandas`: For data manipulation and analysis.
- `numpy`: For numerical operations.
- `scikit-learn`: For machine learning models, preprocessing, and evaluation.
- `xgboost`: For XGBoost model implementation.
- `catboost`: For CatBoost model implementation.
- `imblearn`: For handling imbalanced datasets (e.g., SMOTE).
- `matplotlib`, `seaborn`: For data visualization.
- `pyyaml`: For reading configuration files.
- `jupyterlab`: For running the Jupyter notebooks.

9. Conclusion

This project provides a comprehensive solution for predicting customer churn in the telecommunications industry. By combining rigorous data preprocessing, advanced machine learning models (particularly ensemble methods like Random Forest), and a strong focus on business impact, the solution offers a clear path to significant revenue protection and increased customer lifetime value. The documented ROI of 259% and net benefit of $47,270 highlight the tangible financial advantages of implementing this predictive system. The modular structure and detailed documentation ensure that the project is not only effective but also maintainable and extensible for future enhancements.