Predicting Customer Churn in the Telecommunications Sector

Objective:

This capstone project aims to consolidate and apply the knowledge you have acquired in Python, Machine Learning, and data engineering tools to predict customer churn in the telecommunications sector. The project will require you to build a comprehensive solution that includes data exploration, predictive modeling, and, if possible, model deployment. Your final deliverable will be a professional presentation that showcases your findings and provides actionable insights to the company.

Project Scope:

1. **Understanding the Business Problem:**
    - **Context:** In the highly competitive telecommunications industry, customer retention is crucial. Churn prediction models help companies identify customers at risk of leaving, allowing them to take proactive steps to retain these customers.
    - **Business Goals:** The goal is to predict whether a customer will churn, using the dataset provided, which includes demographic, location, service usage, and customer status data. You will also need to identify the key factors that contribute to churn and suggest strategies to reduce it.
    - **Key Stakeholders:** The primary stakeholders are the marketing and customer service teams, who will use your insights to more effectively target retention efforts.

Note: The suggestions outlined below are recommended approaches, but students have the freedom to choose their own path and explore techniques they find most suitable for the problem.

1. **Data Collection and Initial Processing:**
    - **Dataset Overview:** The dataset consists of five tables: Demographics, Location, Population, Services, and Status. Each table contains various

attributes relevant to customer churn.

- **Data Description:** Understand the structure of each table and the relationships between them. Key variables include CustomerID, Churn Label, Churn Score, CLTV (Customer Lifetime Value), and various demographic and service-related attributes.
- **Data Cleaning:**
    - Address any missing values, inconsistencies, or duplicates in the dataset.
    - Merge tables appropriately to create a unified dataset for analysis.
- **Feature Engineering:**
    - Create additional features that could help improve model performance. For example, combine existing attributes to create new indicators of customer behavior, such as customer engagement scores.
    - Evaluate the impact of feature engineering on model performance through iterative testing.

2. **Exploratory Data Analysis (EDA):**
    - **Univariate Analysis:** Analyze individual features to understand their distributions and identify any anomalies or patterns. Pay special attention to the Churn Label, CLTV, and service-related features.
    - **Bivariate and Multivariate Analysis:** Investigate relationships between features, especially how they relate to churn. Use correlation matrices, heatmaps, and scatter plots to detect significant correlations.
    - **Visualization:** Develop clear and insightful visualizations that communicate the most important findings. Utilize visualization libraries to create interactive dashboards or dynamic plots.
    - **Segment Analysis:** Consider segmenting the customer base (e.g., by contract type or service package) to identify specific groups more likely to churn.
3. **Predictive Modeling:**
    - **Model Selection:**
        - Begin with simple models like Logistic Regression to set a baseline.

    - Progress to more sophisticated models such as Random Forests, Gradient Boosting Machines (e.g., XGBoost), and Neural Networks.
- **Model Tuning:**
    - Fine-tune model parameters using Grid Search or Random Search, with cross-validation to ensure robust performance.
    - Consider balancing techniques like SMOTE if the dataset is imbalanced (e.g., fewer customers who churn).
- **Model Evaluation:**
    - Use metrics like accuracy, precision, recall, F1-score, and AUC-ROC to evaluate model performance. Pay particular attention to precision and recall for the churned customers class.
    - Perform error analysis to understand misclassifications and identify areas for improvement.
- **Model Interpretation:**
    - Use tools such as SHAP or LIME to interpret the predictions of complex models and explain them to non-technical stakeholders.
    - Provide a feature importance analysis to highlight the most influential factors driving churn.

4. **Prescriptive Analytics and Recommendations:**
    - **Insights Derivation:** Based on model predictions, identify key drivers of churn and segment the customer base according to their churn risk.
    - **Scenario Analysis:** Conduct what-if analyses to simulate the impact of different retention strategies on churn rates.
    - **Strategic Recommendations:** Provide targeted recommendations for reducing churn, such as personalized offers for high-risk customers, changes in service plans, or improvements in customer support.
5. **Model Deployment (Optional):**
    - **Deployment Strategy:** Deploy the final model using platforms such as Hugging Face Spaces or any other suitable environment.
6. **Final Presentation and Reporting:**
    - **Presentation Structure:**

    - **Introduction:** Brief overview of the business problem, objectives, and approach.
    - **EDA and Feature Engineering:** Summary of key findings and the rationale behind feature engineering choices.
    - **Modeling and Evaluation:** Discussion of the models developed, performance metrics, and the final model selection.
    - **Recommendations:** Presentation of strategic recommendations based on model insights.
    - **Deployment (Optional):** Demonstration of the deployed model (if applicable) and explanation of its usage.
- **Slide Deck:**
    - Create a polished slide deck that clearly conveys your analysis and recommendations. Ensure the slides are visually appealing and professional.
    - Prepare backup slides with additional data and analysis to address potential questions during the presentation.
- **GitHub Repository:** Maintain a well-organized GitHub repository containing all project files, including code, folders for data, documentation, and the presentation. Ensure the repository is structured to be easily understood by others.

Technologies to be Used:

- **Programming & Data Analysis:** Python (Pandas, NumPy, Scikit-learn, XGBoost, TensorFlow/PyTorch, etc.)
- **Recommended Development Environment:** VS Code
- **Version Control:** Git/GitHub
- **Model Deployment:** Hugging Face Spaces (Optional)
- **Data Visualization:** Plotly, Matplotlib, Seaborn
- **Model Interpretation:** SHAP, LIME

Deliverables:

1. **Jupyter Notebook/Python Scripts:** Comprehensive documentation of your workflow, from EDA to modeling and deployment.

2. **Final Presentation:** A professional presentation with accompanying slides, prepared for a 10-minute delivery.
3. **Backup Slides:** Additional slides with further details for potential deep dives during Q&A.
4. **GitHub Repository:** A complete and well-structured repository with all relevant files and documentation.