Telco Customer Churn Prediction Project Report 📊

1. Introduction ✨

This report outlines the process of building a machine learning model to predict customer churn in a telecommunications company. Customer churn, the rate at which customers stop doing business with an entity, is a critical metric for telco companies as retaining existing customers is often more cost-effective than acquiring new ones. The project encompasses data loading, extensive preprocessing, exploratory data analysis, feature engineering, and model training and evaluation.

2. Dataset Overview 📚

The project utilizes the Telco Customer Dataset.csv, which contains customer information and their churn status.

- **Source:** Telco Customer Dataset.csv
- **Target Variable:** Churn (binary: 'No' for non-churn, 'Yes' for churn).
- **Initial Data Characteristics:**
    - The dataset contains 7043 entries and 21 columns.
    - Columns include customer demographics (gender, SeniorCitizen, Partner, Dependents), service information (PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod), and billing details (tenure, MonthlyCharges, TotalCharges).
    - Initial inspection shows TotalCharges is of object dtype, indicating it likely contains non-numeric values (e.g., spaces or empty strings) that need handling.
    - The customerID column is present, which is an identifier and not a predictive feature.
    - No explicit missing values were initially reported by df.isna().sum(), but the TotalCharges column's data type suggests implicit missing or non-numeric values.
    - **Class Imbalance:** The target variable Churn exhibits a significant imbalance, with approximately 73.46% 'No' (non-churn) and 26.54% 'Yes' (churn). This imbalance is a key challenge that needs to be addressed for effective model training.

3. Methodology 🛠️

3.1 Data Preprocessing & Cleaning 🧹

The raw data underwent several cleaning and transformation steps:

- **Customer ID Removal:** The customerID column was dropped as it serves no predictive purpose.
- **TotalCharges Handling:**
    - The TotalCharges column, initially an object type, was converted to a numeric type. Empty strings (which Pandas reads as objects) were identified and converted to NaN (Not a Number), then imputed with the median of the column.
- **Feature Type Separation:** Columns were separated into numerical (num_cols) and categorical (cat_cols) lists for targeted processing. SeniorCitizen was initially identified as numerical but later removed from numerical columns for skewness analysis, as it's a binary (0/1) indicator.
- **Binary Categorical Mapping:** Several binary categorical columns (gender, Partner, Dependents, PhoneService, PaperlessBilling, Churn) were mapped to numerical representations (0 and 1) for consistency and model compatibility.
- **High Cardinality Categorical Handling:**
    - MultipleLines, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies had 'No internet service' or 'No phone service' values, which were mapped to 'No' to consolidate categories.
- **One-Hot Encoding:** Remaining categorical features were identified for one-hot encoding, converting them into numerical format suitable for machine learning models.

3.2 Exploratory Data Analysis (EDA) 📊

- **Target Distribution:** A count plot confirmed the class imbalance in the Churn variable, highlighting the need for resampling techniques.
- **Numerical Feature Distribution:** Box plots were used to visualize the distribution of numerical features (tenure, MonthlyCharges, TotalCharges) and identify potential outliers. Skewness was also calculated for these columns.
- **Categorical Feature Distribution:** Count plots were generated for all categorical features to understand their value distributions.
- **Relationship with Target:**

    - Count plots and cross-tabulations were used to analyze the relationship between categorical features and Churn.
    - Box plots were used to visualize the relationship between numerical features and Churn.

3.3 Feature Transformation & Scaling 🔄

- **Skewness Transformation:** np.log1p (logarithmic transformation) was applied to tenure, MonthlyCharges, and TotalCharges to reduce skewness and normalize their distributions.
- **Standard Scaling:** All numerical features were scaled using StandardScaler to ensure they contribute equally to the model, preventing features with larger values from dominating.
- **Categorical Encoding:** One-hot encoding was applied to all remaining categorical features, transforming them into a numerical format.

3.4 Data Splitting & Resampling 🧩

- The dataset was split into **training (80%) and test (20%) sets** using train_test_split.
- To address the significant class imbalance in the Churn target, **SMOTE (Synthetic Minority Over-sampling Technique)** was applied to the training data. This technique oversamples the minority class ('Yes' churn) by creating synthetic samples, resulting in a balanced training set for model learning.

3.5 Model Training & Evaluation 🧠

A Logistic Regression model was chosen for prediction, and its performance was thoroughly evaluated.

- **Model:** Logistic Regression (LogisticRegression)
- **Hyperparameters:** The model was initialized with C = 10, penalty = 'l1', solver = 'liblinear', and a class_weight dictionary {0: 1, 1: 1.5}. The class_weight parameter explicitly addresses the class imbalance by giving more importance to the minority class (churn).
- **Training:** The model was trained on the SMOTE-resampled training data.
- **Evaluation Metrics:**
    - **Accuracy Score:** Overall correctness of predictions.

- **Confusion Matrix:** Provides a detailed breakdown of true positives, true negatives, false positives, and false negatives.
- **Classification Report:** Presents precision, recall, and F1-score for each class, which are crucial for imbalanced datasets.
- **ROC Curve and AUC:** The Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) were used to assess the model's ability to distinguish between churn and non-churn classes across various thresholds.
- **Precision-Recall vs. Threshold Plot:** A plot was generated to visualize the trade-off between precision and recall at different probability thresholds, allowing for the selection of an optimal threshold based on business needs.

4. Results and Evaluation 📈

4.1 Logistic Regression Model Performance

The Logistic Regression model, after training on SMOTE-resampled data and using specific class weights, was evaluated on the unseen test set.

- **Accuracy on Final Test Set:** [Insert Accuracy Score from your output, e.g., 0.78]
- **Confusion Matrix of Final Test Set:**
- [[TN FP]
- [FN TP]]

[Insert Confusion Matrix from your output, e.g., [[1135 244] [140 242]]]

- **True Negatives (TN):** [Value] (Correctly predicted non-churners)
- **False Positives (FP):** [Value] (Incorrectly predicted churners - Type I error)
- **False Negatives (FN):** [Value] (Incorrectly predicted non-churners - Type II error, missed churners)
- **True Positives (TP):** [Value] (Correctly predicted churners)

- **Classification Report on Final Set:**
- [Insert Classification Report from your output]

    - **Precision (for Churn=Yes):** [Value] (Out of all predicted churners, how many were actually churners?)
    - **Recall (for Churn=Yes):** [Value] (Out of all actual churners, how many did the model correctly identify?)
    - **F1-Score (for Churn=Yes):** [Value] (Harmonic mean of precision and recall, good for imbalance)
- **ROC Curve and AUC:**
    - The ROC curve visually represents the trade-off between the true positive rate (recall) and the false positive rate.
    - The AUC score quantifies the overall performance of the classifier, indicating its ability to distinguish between classes. [Insert AUC score from your output/plot].
- **Precision vs. Recall vs. Threshold Plot:**
    - This plot is critical for understanding how different probability thresholds impact the balance between precision and recall for the churn class. It allows for a strategic choice of threshold based on whether minimizing false positives (higher precision) or minimizing false negatives (higher recall) is more important for the business. The plot shows a 'best threshold' selected, which is a valuable insight.

5. Conclusion and Future Work 💡

5.1 Conclusion 🎉

The Telco Customer Churn Prediction project successfully developed a Logistic Regression model capable of identifying potential customer churn. Through meticulous data preprocessing, handling of the TotalCharges column, strategic feature engineering, and addressing class imbalance with SMOTE and class weighting, the model achieved a balanced performance on the test set. The detailed evaluation using accuracy, confusion matrix, classification report, ROC curve, and precision-recall threshold analysis provides a comprehensive understanding of the model's strengths and areas for improvement. The ability to tune the prediction threshold based on business priorities (e.g., prioritizing recall to capture more churners for intervention) is a significant outcome.

5.2 Future Enhancements 🚀

- **Explore More Models:** While Logistic Regression is a good baseline, investigate other robust classification algorithms like Gradient Boosting (XGBoost, LightGBM), Random Forests, or even simple Neural Networks, and perform hyperparameter tuning for them.

- **Advanced Feature Engineering:**
    - Create interaction terms between features (e.g., tenure * MonthlyCharges).
    - Derive new features from existing ones that might better capture customer behavior (e.g., average monthly charge per tenure).
- **Ensemble Methods:** Experiment with ensemble techniques such as stacking or blending to combine the predictions of multiple models for potentially higher accuracy and robustness.
- **More Sophisticated Imbalance Handling:** Explore other advanced resampling techniques (e.g., ADASYN, Borderline-SMOTE) or different approaches like cost-sensitive learning, which directly incorporates the cost of misclassification into the model's objective function.
- **Explainable AI (XAI):** Implement techniques like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) to understand which features are most influential in predicting churn for individual customers, providing actionable insights for marketing and customer retention teams.
- **Hyperparameter Optimization:** Implement more systematic hyperparameter optimization strategies like Randomized Search or Bayesian Optimization, especially if exploring more complex models.
- **Deployment:** Develop a simple web application or API to demonstrate the model's real-time prediction capabilities.