Code Explanation: Maternal Mortality Risk Prediction Model

This document explains the Python script for building and evaluating a machine learning model to predict maternal mortality risk using a LightGBM classifier. The script processes a dataset, applies preprocessing, handles class imbalance, performs cross-validation, evaluates model performance, and analyzes feature importance using SHAP. The best results from the model evaluation are also included.

---

1. Imports and Setup

The script begins by importing necessary libraries and suppressing warnings to reduce clutter in the output.

- **Libraries**:
  - `pandas` and `numpy` for data manipulation and numerical operations.
  - `lightgbm` for the gradient boosting model.
  - `sklearn` modules for model evaluation (`roc_auc_score`, `f1_score`, etc.), data splitting (`train_test_split`, `StratifiedKFold`), and metrics calculation (`precision_recall_curve`, `auc`).
  - `imblearn.combine.SMOTEENN` for handling class imbalance.
  - `category_encoders.TargetEncoder` for encoding categorical features.
  - `shap` for model interpretability.
  - `matplotlib.pyplot` for plotting SHAP visualizations.
  - `pyarrow.parquet` for reading Parquet files.
- **Warnings**: `warnings.filterwarnings('ignore', category=UserWarning)` suppresses user warnings to keep the output clean.
- **SMOTEENN Check**: Verifies that `SMOTEENN` is available, raising an error if not installed.

---

2. Data Preprocessing (`prepare_data_for_targets`)

The `prepare_data_for_targets` function loads and preprocesses data from a Parquet file in batches to handle large datasets efficiently.

Key Steps:

- **Batch Processing**:
  - Reads the Parquet file (`telangana_data_with_features_and_targets (1).parquet`) using `pyarrow.parquet`.
  - Processes data in chunks (`batch_size=10000`) to manage memory usage.
- **Column Definitions**:
  - **Required Columns**: Ensures `MOTHER_ID` and `GRAVIDA` are present.
  - **Numeric Columns**: Includes features like `GRAVIDA`, `PARITY`, `HEIGHT`, `BMI`, etc.
  - **Flag Columns**: Binary indicators like `age_adolescent`, `hypertension`, etc.
  - **Categorical Columns**: Features like `FACILITY_TYPE`, `BLOOD_GRP`, `SYS_DISEASE`.
- **Preprocessing**:
  - Converts numeric columns to numeric type, handling errors by coercing to `NaN`.
  - Maps flag columns (`Y`, `YES`, etc. → 1; `N`, `NO`, etc. → 0) and imputes `NaN` with 0.
  - Imputes missing numeric values with the median and categorical values with the mode.
  - Limits `SYS_DISEASE` to the top 10 categories, labeling others as `Other`.
  - Creates interaction features: `anemia_severe_systolic_bp` and `hypertension_hemoglobin`.
- **Output**: Returns a preprocessed DataFrame with no missing values.

---

3. Stratified Sampling (`create_stratified_sample`)

The `create_stratified_sample` function creates a balanced sample of the dataset, ensuring all maternal mortality cases are included due to their rarity.

Key Steps:

- **Input Validation**: Checks if the target column (`maternal_mortality_risk`) exists.
- **Positive Case Handling**:
  - Identifies all positive cases (`maternal_mortality_risk == 1`).
  - If fewer than `min_positive` (1000) positive cases are found, oversamples with replacement to meet this threshold.
- **Sampling**:

    - Includes all positive cases (1,377 cases) and samples negative cases to reach the desired `sample_size` (1,000,000), resulting in 998,623 negative cases.
- **Output**: Returns a stratified sample DataFrame with 1,377 positive and 998,623 negative cases.

---

4. Main Script Execution

The main script orchestrates data loading, preprocessing, model training, evaluation, and interpretability analysis.

Steps:

1. **Data Loading**:
    - Loads the dataset with 4,028,194 negative and 1,377 positive cases for `maternal_mortality_risk`.
    - Calls `prepare_data_for_targets` to preprocess the dataset.
2. **Diagnostic Checks**:
    - Prints column names (e.g., `ANC_ID`, `MOTHER_ID`, `GRAVIDA`, etc.) and the distribution of `maternal_mortality_risk`.
    - Confirms positive cases exist.
3. **Stratified Sampling**:
    - Creates a sample with 1,000,000 records (1,377 positive, 998,623 negative).
4. **Feature Selection**:
    - Uses 21 numeric features (e.g., `GRAVIDA`, `HEMOGLOBIN_mean`, `BMI`), 16 flag features (e.g., `age_adolescent`, `hypertension`), and 3 categorical features (`FACILITY_TYPE`, `BLOOD_GRP`, `SYS_DISEASE`).
5. **Data Preparation**:
    - Extracts features (`X`) and target (`y`) from the sampled DataFrame.
    - Confirms no `NaN` values before and after encoding.
    - Uses `TargetEncoder` to encode categorical features.
    - Splits data into training (80%) and test (20%) sets with stratification.

---

5. Cross-Validation and Model Training

The script uses stratified k-fold cross-validation (5 folds) to train and evaluate a LightGBM classifier.

Key Steps:

- **Setup**:
  - Initializes `StratifiedKFold` with 5 splits.
  - Calculates `scale_pos_weight` as 1,087.43 (1.5 × negative-to-positive ratio).
  - Sets LightGBM parameters: binary objective, AUC metric, 500 estimators, max depth 7, etc.
- **Cross-Validation Loop**:
  - For each fold:
    - Splits training data (639,118 negative, ~882 positive) and validation data (159,780 negative, ~220 positive).
    - Applies `SMOTEENN` (sampling strategy = 0.1), resulting in ~631,000 negative and ~61,000 positive cases.
    - Trains a LightGBM model with early stopping (50 rounds).
    - Evaluates metrics (F1, accuracy, precision, recall, PR-AUC) for thresholds (0.1, 0.2, 0.3, 0.4).
- **Metrics**:
  - Computes ROC-AUC and PR-AUC per fold.
  - Prints confusion matrices and metrics for each threshold.

---

6. Test Set Evaluation

The best model (from Fold 2, average F1 score: 0.0031) is evaluated on the test set.

Key Steps:

- **Test Set Metrics**:
  - Predicts probabilities on the test set (199,725 negative, 275 positive cases).
  - Computes ROC-AUC (0.6690), PR-AUC (0.0033), and metrics for thresholds (0.1, 0.2, 0.3, 0.4).
  - Best threshold: 0.4 with F1 score of 0.0036.
- **Output**: Prints test set metrics and confusion matrices.

---

7. SHAP Analysis

The script uses SHAP to interpret the best model's predictions.

Key Steps:

- Samples 1,000 test set instances.
- Creates a `TreeExplainer` for the LightGBM model.
- Computes SHAP values for the positive class.
- Generates and saves:
    - **Summary Plot**: `shap_summary_plot_maternal.png`.
    - **Bar Plot**: `shap_importance_bar_maternal.png`.
- Outputs a feature importance DataFrame, with top features:
    - `WEIGHT_child_min` (0.950696)
    - `inadequate_weight_gain` (0.652248)
    - `HEMOGLOBIN_mean` (0.526082)
    - `anemia_mild` (0.470680)
    - `BLOOD_GRP` (0.331091)

---

8. Best Results

The best results from the model evaluation are summarized below:

Cross-Validation Mean Metrics:

- **AUC**: 0.6303 ± 0.0197
- **Threshold 0.1**:
    - F1 Score: 0.0028 ± 0.0000
    - Accuracy: 0.0562 ± 0.0241
    - Precision: 0.0014 ± 0.0000
    - Recall: 0.9782 ± 0.0178
    - PR-AUC: 0.0030 ± 0.0005
- **Threshold 0.2**:
    - F1 Score: 0.0024 ± 0.0012
    - Accuracy: 0.3115 ± 0.3446
    - Precision: 0.0012 ± 0.0006
    - Recall: 0.7474 ± 0.3745
    - PR-AUC: 0.0030 ± 0.0005
- **Threshold 0.3**:
    - F1 Score: 0.0025 ± 0.0013
    - Accuracy: 0.3701 ± 0.3161
    - Precision: 0.0013 ± 0.0006
    - Recall: 0.7220 ± 0.3622
    - PR-AUC: 0.0030 ± 0.0005

- **Threshold 0.4**:
    - F1 Score: 0.0026 ± 0.0013
    - Accuracy: 0.4286 ± 0.2877
    - Precision: 0.0013 ± 0.0007
    - Recall: 0.6794 ± 0.3421
    - PR-AUC: 0.0030 ± 0.0005

Test Set Metrics (Best Model from Fold 2):

- **AUC**: 0.6690
- **PR-AUC**: 0.0033
- **Threshold 0.1**:
    - F1: 0.0029
    - Accuracy: 0.0898
    - Precision: 0.0015
    - Recall: 0.9636
    - Confusion Matrix: [[17691, 182034], [10, 265]]
- **Threshold 0.2**:
    - F1: 0.0031
    - Accuracy: 0.1731
    - Precision: 0.0015
    - Recall: 0.9236
    - Confusion Matrix: [[34369, 165356], [21, 254]]
- **Threshold 0.3**:
    - F1: 0.0033
    - Accuracy: 0.2511
    - Precision: 0.0016
    - Recall: 0.8945
    - Confusion Matrix: [[49974, 149751], [29, 246]]
- **Threshold 0.4** (Best):
    - F1: 0.0036
    - Accuracy: 0.3295
    - Precision: 0.0018
    - Recall: 0.8691
    - Confusion Matrix: [[65655, 134070], [36, 239]]

Best Threshold:

- **Threshold**: 0.4
- **F1 Score**: 0.0036

---

9. Error Handling and Robustness

- **SMOTEENN Availability**: Verifies `SMOTEENN` and provides installation instructions if missing.
- **Missing Columns**: Validates required columns and raises errors if absent.
- **No Positive Cases**: Ensures positive cases exist in the target.
- **Single-Class Folds**: Skips folds with one class and warns the user.
- **NaN Handling**: Repeatedly checks for and imputes `NaN` values.
- **Data Validation**: Ensures features and target columns are properly formatted.

---

10. Output

- **Console Output**:
  - Data diagnostics (class distribution, column names).
  - Fold-wise metrics and confusion matrices.
  - Cross-validation summary.
  - Test set metrics and best threshold.
  - SHAP feature importance table.
- **Files**:
  - Saves SHAP plots as `shap_summary_plot_maternal.png` and `shap_importance_bar_maternal.png`.

---

11. Key Features of the Code

- **Efficient Data Handling**: Processes large datasets in batches.
- **Class Imbalance Handling**: Uses `SMOTEENN` and stratified sampling.
- **Robust Preprocessing**: Handles missing values and encodes categorical features.
- **Comprehensive Evaluation**: Evaluates multiple thresholds and metrics.
- **Interpretability**: Provides SHAP-based feature importance.
- **Error Handling**: Ensures robustness with extensive checks.

This script is designed for predicting maternal mortality risk, addressing class imbalance, and providing interpretable results in a medical context.