**Model Building Documentation**

Overview

This script automates the process of loading, preprocessing, training, and evaluating machine learning models for churn prediction. The workflow includes:

1. Connecting to a PostgreSQL database to retrieve data.
2. Preprocessing the data, including scaling and handling class imbalance.
3. Training multiple models using hyperparameter tuning via Grid Search.
4. Evaluating the best-performing models.
5. Logging results and saving the best model.

Script Components

1. Import Required Libraries

The script imports essential libraries for:

* Data handling (pandas)
* Machine learning (scikit-learn, imbalanced-learn)
* Database connection (psycopg2)
* Model tracking and storage (joblib, mlflow)

1. Database Connection

* The script connects to a PostgreSQL database using psycopg2.
* Retrieves data from the specified table and handles connection errors.

1. Data Preprocessing

* Removes missing values.
* Scales the numerical feature all\_charges using StandardScaler.
* Defines feature set (X) and target variable (y).
* Applies SMOTE to handle class imbalance.
* Splits data into training and testing sets.
* Standardizes features.

1. Model Training

* Defines multiple machine learning models (Random Forest, Gradient Boosting, KNN, Logistic Regression) with hyperparameter grids.
* Uses GridSearchCV with Stratified K-Fold cross-validation to find the best hyperparameters.
* Logs training details using MLflow.

1. Model Evaluation

* Predicts on the test set using the best models.
* Computes evaluation metrics:
  + Accuracy
  + ROC AUC
  + F1 Score
  + Precision
  + Recall
* Stores results in a CSV file.

1. Model Storage

* Saves the best-performing model using joblib.
* Stores the performance report as a CSV file in a timestamped directory.

Expected Outcome

Upon execution:

* Data is preprocessed and split.
* Models are trained and the best hyperparameters are selected.
* Evaluation metrics are generated and stored.
* The best model is saved for future use.

Troubleshooting

* Database Connection Issues: Verify credentials and database availability.
* Model Training Errors: Check for data inconsistencies.
* Performance Metrics Too Low: Adjust hyperparameters or explore additional feature engineering.