Name: **Sushain Devi**

ML Engineer intern - Assignment

**Report: Loan Default Prediction Model**

**1. Introduction**

This project aims to predict loan defaults for borrowers using data from the Branch mobile app. The model is built using various features, including user financial information and GPS data, to determine whether a borrower will repay their loan in full. The goal is to implement a solution that can assist Branch in making informed lending decisions.

**2. Data Exploration and Feature Engineering**

**Data Access**

The dataset consists of three primary tables stored in a PostgreSQL database:

* **loan\_outcomes**: Contains loan application dates and repayment status (repaid or defaulted).
* **gps\_fixes**: Contains GPS-related data for users, recorded when the user opens the Branch app.
* **user\_attributes**: Contains additional user information, such as age and incoming cash in the last 30 days.

I am connecting to the database and extracting the data using the pandas library and SQLAlchemy. Upon retrieving the data, an initial inspection is done to check for missing values and basic descriptive statistics.

**Data Merging**

The data from the three tables is merged into a single DataFrame using the user\_id as the key. This enables combining user-specific information, GPS data, and loan outcomes.

**Feature Engineering**

Several features are engineered from the available data:

* **Loan Outcome**: Converted the binary "yes"/"no" outcome to numerical values (1 and 0).
* **GPS Features**: Derived various statistics from the GPS data, including:
  + The count of GPS fixes per user.
  + The mean and variance of latitude and longitude.
  + The movement radius (estimated from the bounding box of GPS data).
  + The user's peak activity hour based on GPS fix timestamps.

**Missing Value Handling**

Missing values in the features are handled by imputing the GPS-related data with zeros and using the median for numerical values like age and cash incoming.

**3. Model Training**

**Methods Used for Model Training**

For this problem, I explored several machine learning models, including:

* **Logistic Regression**: A basic model for binary classification, but struggled to handle non-linear relationships between features.
* **Gradient Boosting Classifier**: A powerful model for classification tasks, but it required more tuning to achieve optimal results.
* **Random Forest Classifier**: A robust model for binary classification that handles feature interactions well and provides feature importance scores. This was chosen as the final model due to its ability to handle a mix of numerical and categorical data effectively.

I used **Random Forest Classifier** as it performed well for this particular problem, handling the mixed data types and large number of features efficiently. It also provided an interpretable feature importance ranking, helping to identify the most influential features.

**Model Performance**

I evaluated the model's performance using cross-validation with ROC-AUC as the primary metric. The results showed good classification performance despite the limited dataset.

**Hyperparameter Tuning**

To improve the model's performance, I performed hyperparameter tuning using **GridSearchCV**, optimizing parameters such as n\_estimators, max\_depth, and min\_samples\_split. The best combination of parameters was used for training the final model.

**Model Evaluation**

After training, I used various evaluation metrics:

* **ROC-AUC Curve**: To assess the classifier's ability to distinguish between the two classes (default and non-default).
* **Confusion Matrix**: To visualize the performance of the classifier in terms of false positives and false negatives.
* **Classification Report**: To provide precision, recall, and F1-score.

**4. Model Deployment**

**API for Prediction**

I exposed the trained model through a simple API using **FastAPI**. The API accepts a user's data, applies necessary preprocessing steps, and returns the predicted loan default probability and a confidence score.

The key endpoints are:

* /predict: For making loan default predictions.
* /model-info: For retrieving information about the model, including the feature importance and parameters.
* /health: For performing a basic health check of the API.

**Code Files**

The complete code for this project, including data analysis, model training, and API deployment, is saved in the following GitHub repository:

[**GitHub Repository**](https://github.com/SushainDevi/intern-Assignment-BRANCH)

**5. Future Improvements**

The model's performance could be further improved by:

* **Increasing the dataset size**: A larger dataset would help train a more accurate model.
* **Additional features**: Exploring more complex features from GPS data (e.g., trajectory analysis) and user behavior could improve predictive performance.
* **Model Ensemble**: Using an ensemble of models or stacking multiple models to combine their predictions could increase accuracy.
* **Real-time Data**: Incorporating real-time user data for dynamic predictions might improve model performance in production environments.

**6. Conclusion**

This project demonstrates how data exploration, feature engineering, and machine learning can be applied to solve a real-world problem like predicting loan defaults. The **Random Forest Classifier** was chosen as the optimal model for this task, and its performance was evaluated through various metrics. The model has been successfully deployed through a FastAPI endpoint for predictions.