**Detailed Report: Product Purchase Prediction using Random Forest Regression**

**1. Introduction**

The goal of this analysis is to predict the purchase amount for each product based on a range of features available in the training dataset. The prediction model is then used to generate predictions on the test dataset and create a submission file that follows a given format. A Random Forest Regressor is used as the primary modeling technique.

**2. Data Preprocessing**

**2.1 Data Loading**

* **Datasets:**
  + **Train Data:** Loaded from train.csv.
  + **Test Data:** Loaded from test.csv.
  + **Sample Submission:** Loaded from Samle\_Submission.csv (to verify the required schema for the final submission).
* **Initial Inspection:**
  + The columns in the train and test datasets are printed to verify consistency.
  + The train dataset contains:  
    User\_ID, Product\_ID, Gender, Age, Occupation, City\_Category, Stay\_In\_Current\_City\_Years, Marital\_Status, Product\_Category\_1, Product\_Category\_2, Product\_Category\_3, and Purchase.
  + The test dataset contains the same features except for Purchase.

**2.2 Missing Value Analysis**

* **Train Data:**
  + Missing values are primarily observed in:
    - Product\_Category\_2 (173,638 missing values)
    - Product\_Category\_3 (383,247 missing values)
* **Test Data:**
  + Missing values are present in:
    - Product\_Category\_2 (72,344 missing values)
    - Product\_Category\_3 (162,562 missing values)

These counts indicate that several product category details are missing. A simple imputation strategy (filling with 0) is used to handle these missing values.

**2.3 Combining and Transforming the Data**

* **Concatenation for Consistent Transformations:**
  + To ensure that the same transformations are applied to both training and test sets, the datasets are combined into a single DataFrame.
  + A flag column is\_train is added (1 for training data and 0 for test data).
  + A temporary placeholder for Purchase is set in the test set to align columns during the merge.
* **Categorical Encoding:**
  + **Encoded Columns:**  
    The following categorical features are label encoded using LabelEncoder:
    - Gender
    - Age
    - City\_Category
    - Stay\_In\_Current\_City\_Years
  + **ID Columns:**  
    Both User\_ID and Product\_ID are label encoded to allow the model to potentially capture any patterns associated with these identifiers.
* **Handling Missing Values in Product Categories:**
  + The missing values in Product\_Category\_2 and Product\_Category\_3 are filled with 0.
* **Splitting the Data Back:**
  + After transformation, the combined DataFrame is split back into:
    - **train\_cleaned:** Rows where is\_train equals 1.
    - **test\_cleaned:** Rows where is\_train equals 0.

**3. Exploratory Data Analysis (EDA)**

**3.1 Structural Overview**

* **Columns Verification:**
  + The columns for both train and test datasets are printed to confirm that they match the expected schema.
* **Missing Value Summary:**
  + A summary of missing values is printed for both datasets. This confirms that missing data are confined to the product category columns and that other important features (e.g., User\_ID, Product\_ID, and Purchase) are complete.

**3.2 Observations**

* **Missing Data Handling:**
  + The high number of missing values in the product category fields indicates incomplete data collection for these features. Filling them with 0 is a straightforward imputation strategy to allow modeling to proceed.
* **Potential for Further Analysis:**
  + While the notebook does not include additional visualizations (like histograms or box plots), further EDA could include examining the distribution of numeric features and analyzing correlations between variables, which might inspire further feature engineering.

**4. Modeling Techniques**

**4.1 Train/Validation Split**

* **Local Evaluation:**
  + The cleaned training data is split into a training set and a validation set using an 80/20 ratio. This split allows for an estimation of model performance before final predictions on the test set.

**4.2 Model Training**

* **Algorithm:**
  + A **RandomForestRegressor** is chosen for its robustness and ability to capture non-linear relationships.
  + Model hyperparameters used include:
    - n\_estimators = 100
    - random\_state = 42 (ensuring reproducibility)
* **Training Process:**
  + The model is trained using the training portion of the data.
  + Feature set X consists of all columns except Purchase and is\_train.
  + The target y is the Purchase column.

**4.3 Model Evaluation**

* **Validation Predictions:**
  + The trained model makes predictions on the validation set.
* **Error Metric:**
  + The performance is evaluated using the Root Mean Squared Error (RMSE), which is calculated as the square root of the mean squared error (MSE).
  + The reported **Validation RMSE is approximately 2753.88**, indicating the average deviation of predictions from the true purchase amounts.

**5. Inference and Submission**

**5.1 Test Set Predictions**

* **Inference:**
  + The model is applied to the test set (after dropping Purchase and is\_train columns) to generate predictions for each record.

**5.2 Submission File Creation**

* **Submission Schema:**
  + The final submission file is constructed by combining:
    - User\_ID
    - Product\_ID
    - Predicted Purchase values
  + This matches the format of the provided sample submission file.
* **Output:**
  + The submission file is saved as my\_submission.csv.
  + A confirmation message is printed indicating that the submission file was created successfully.

**6. Conclusion and Future Work**

**6.1 Summary**

* **Data Preprocessing:**
  + Data from both train and test datasets were loaded and combined for consistent transformations.
  + Missing values in product category columns were imputed with 0.
  + Categorical features were encoded using LabelEncoder, including high-cardinality ID fields.
* **Modeling:**
  + The RandomForestRegressor was trained on the preprocessed data.
  + A local train/validation split was used to estimate model performance, achieving a validation RMSE of approximately 2753.88.
* **Submission:**
  + Predictions on the test set were generated and saved in a submission file that adheres to the sample format.

**6.2 Future Directions**

* **Hyperparameter Tuning:**
  + Future iterations could involve tuning the model parameters (e.g., increasing the number of trees, adjusting maximum tree depth) using grid search or random search techniques to potentially reduce the RMSE.
* **Enhanced Feature Engineering:**
  + Additional features could be engineered, such as aggregating categorical features or creating interaction terms between variables, to improve model performance.
* **Model Experimentation:**
  + Exploring other regression models (e.g., Gradient Boosting, XGBoost, or LightGBM) might yield better predictive accuracy.
* **Deeper EDA:**
  + A more comprehensive exploratory analysis including visualizations (histograms, scatter plots, correlation matrices) could further enhance understanding of the data and inform subsequent modeling choices.