**Performance Report – Raw vs Preprocessed Dataset (70:30 and 80:20 Split)**

**Goal:**

I wanted to check how a machine learning model performs on a raw dataset compared to a cleaned/preprocessed version. Also tried two different data splits to see how training size and validation size affect performance.

**What I Thought Before Starting:**

* The dataset had missing values and some text columns, so I expected the raw version to perform worse.
* Preprocessing (filling missing values, encoding categories, etc.) would probably give the model more complete data to learn from.
* I wasn’t sure whether 70:30 or 80:20 would be better. More training data could help, but less validation/test data might make it harder to measure real performance.

**Steps I Followed:**

* Dropped the Legacy\_Customer\_ID column because it's just an identifier.
* Chose Random Forest Classifier since it works well without needing much tuning and handles both numbers and categories.
* For the **raw dataset**, I just dropped any rows that had missing values.
* For the **preprocessed version**, I:
  + Filled missing numeric values with the column mean.
  + Filled missing categorical values with the most frequent category.
  + Used OneHotEncoding for all categorical variables.
  + Removed the Customer\_Feedback column since it was just raw text and I wasn’t doing NLP yet.

**Results on Raw Dataset:**

**70:30 Split**:

* After dropping missing values, only ~570 rows were left to train on.
* Validation accuracy was ~58%, and F1-score was very low (~0.21).
* Test accuracy was ~57%, F1 even lower (~0.17).
* So, it wasn't doing a great job, especially with predicting the minority class.

**80:20 Split**:

* Got a little more training data, but less for validation/testing.
* Validation accuracy dropped to ~55%, but test accuracy went up to ~60%.
* Test F1-score was better (~0.31), so it picked up the positive class a bit better.
* But it felt like it was slightly overfitting since validation was weaker.

**Results on Preprocessed Dataset:**

**70:30 Split**:

* Used the full 1000 rows since I filled in the missing values.
* Validation accuracy stayed around ~55%, but test accuracy improved a lot to **64%**.
* Test F1-score also went up to **0.34**, which was the best out of all.
* So preprocessing definitely helped.

**80:20 Split**:

* Validation accuracy went up to 58%, F1 was 0.3.
* But test accuracy dropped to ~59% and F1 to 0.25.
* So maybe it learned well from training but didn’t generalize as much.

**What I Learned:**

* Dropping rows with missing values cuts down the training data and hurts model performance.
* Preprocessing the data (filling in gaps and encoding properly) gave better results across the board.
* F1-score is more useful than accuracy when the dataset might be unbalanced. Accuracy looked okay sometimes, but F1 showed that the model wasn’t predicting class 1 very well.
* 70:30 split seems more balanced in terms of training and evaluation. 80:20 had more training data but sometimes didn’t generalize well to test data.