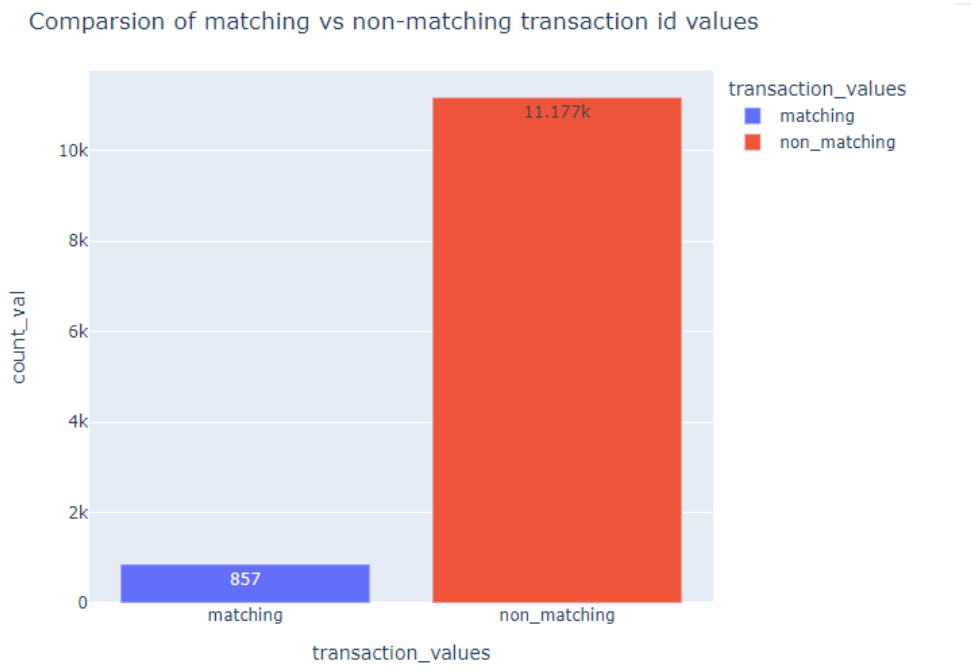
Problem Statement

The problem statement is regarding matching the receipt images associated with the transaction id from the tide app. Now, this data is given by an external vendor who usually contains different features such as DateMappingMatch, ShortNameMatch, DescriptionMatch and other different features based upon which filtering is done and subsequent transaction ids are generated which can be subsequent matches to the original transaction. Here we have to come up with a Machine learning Model which can basically distinguish different transaction ids generated and arrange them in order of the probabilities of matching with the original transaction id.

Target Variable

Let us see the distribution of the target variable

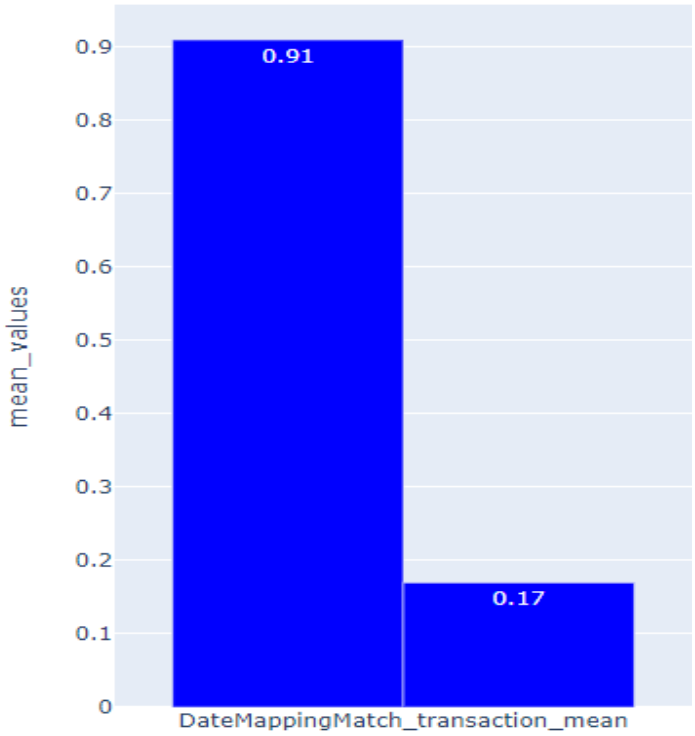


Here we can clearly see the dataset is highly unbalanced with only 7% of the values belonging to the positive class.

Basic EDA

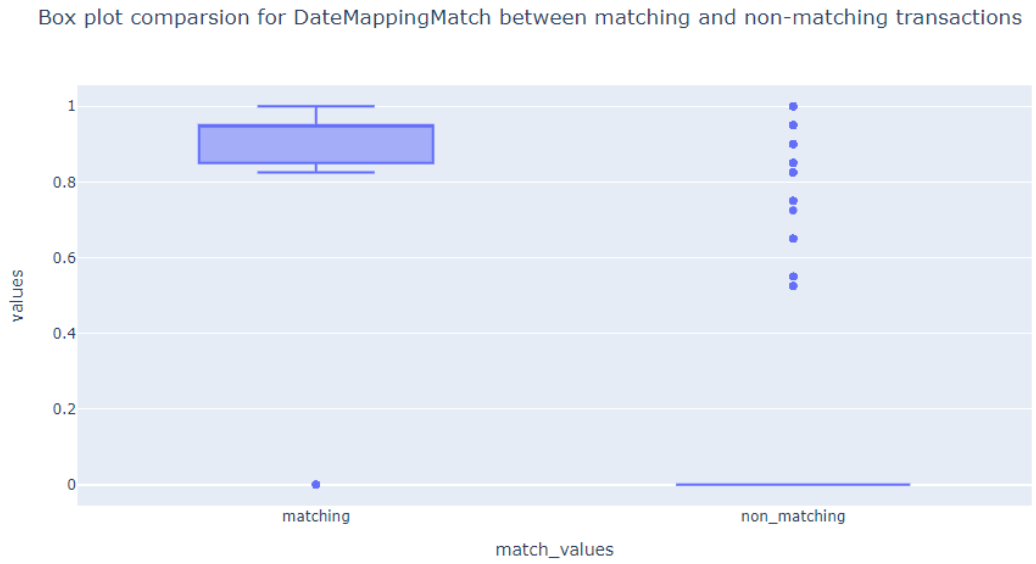
Data Analysis is giving us some insights which definitely help us to understand the features which might be important for segregation purposes. Here I am attaching some of the plots with explanations for that. A detailed explanation is present in the Jupyter Notebook

1) The first part is related to the average values for different features for matching vs non-matching transaction ids(DateMappingMatch, ShortNameDescription..etc)



Here we can clearly see a difference between the average values which are present for ids labeled as 0/1.This can definitely be an important feature for the model which can help us to classify the transactions into positive and negative class

2) Let us visualize the box plot for DateMappingMatch for labels 0 and 1



If we observe for most of the non-matching transactions DateMappingMatch values are also zero with very few values that are greater than zero. For non-matching transaction values which are greater than zero even when it is as high as Matching transaction values but still labeled as zero should be featured in such a way that even when those values are high model should be able to differentiate.

3) The third step is to analyze the correlation plot between different numerical variables

	DateMappingMatch	AmountMappingMatch	DescriptionMatch	DifferentPredictedTime	TimeMappingMatch
DateMappingMatch	1.00	-0.01	0.15	-0.19	0.19
AmountMappingMatch	-0.01	1.00	-0.01	-0.01	0.01
DescriptionMatch	0.15	-0.01	1.00	-0.09	0.09
DifferentPredictedTime	-0.19	-0.01	-0.09	1.00	-0.99
TimeMappingMatch	0.19	0.01	0.09	-0.99	1.00
PredictedNameMatch	0.18	-0.03	0.27	-0.13	0.13
ShortNameMatch	0.20	0.03	0.10	-0.16	0.16
DifferentPredictedDate	-0.99	0.01	-0.15	0.18	-0.18
PredictedAmountMatch	0.02	0.29	0.00	0.01	-0.01
PredictedTimeCloseMatch	0.16	0.02	0.08	-0.41	0.40

From the correlation plot, we can clearly see that DateMappingMatch, DifferentPredictedDate, and TimeMappingMatch, DifferentPredictedTime are highly correlated with a negative magnitude. We definitely will be processing to avoid multicollinearity issues

Performance Metric

The metrics which are used for evaluation are F1-Score, Auc-Roc Score along with Precision and recall. All these parameters help us to understand the trade-offs which happen when we are considering different models.

Experimental Models

Model	F1-Score	Auc-Score	Precision	Recall
Xgboost	0.69	0.92	0.87	0.57
LGBM	0.69	0.89	0.81	0.59
RF-Regressor	0.68	0.92	0.87	0.56
Logistic-Regression	0.67	0.89	0.82	0.56

Precision- It tells us the fact that whenever we are predicting that a particular transaction belongs to a positive belongs how accurate we are in doing that. Xgboost and Rf-Regressor are two models which is giving us a precision of 87%. Even if we observe the Auc score it comes out to be 92%(which is due to the fact that we are able to predict positive class labels with higher accuracy)

Recall-It gives us information that how many points we are able to classify correctly out of total positive points. LightGBM model is having higher recall as compared to the other models(by approx. 2-3%) but has lower precision than the other 3 models with AUC score same as the Logistic Regression Model.

F1-Score-This is a very useful metric when we want to have a balance between precision and recall. It is the harmonic mean of Precision and Recall(this fact helps us it to maintain the balance between precision and recall). So considering F1-Score and AUC Score we can definitely go with the Xgboost model as it maintains the balance between both the two discussed metrics

Conclusion

A total of 4 different models were trained out of which Xgboost is giving a slightly better performance as compared to the other 3 models. AS we saw we are able to achieve precision near about 90% but recall is lower as points belonging are present in the higher majority which makes the class severely imbalanced.

To improve the overall F1-Score we definitely need more information about the data so that we can come up with solid engineered features which will help to reduce the false negatives.

Although the parameters are tuned using Optuna but still we can tune them in a more rigorous way and come with an ensembling approach that can help to create more advanced models which average out the error made by the different individual models

Another thing that can be tried out is the class weight feature which different boosting/bagging models have which helps to deal with class imbalances as it imposes more penalty when an positive class point is misclassified