Receipt Matching Data Science Challenge

Problem Statement

Goal is to match a single receipt to the correct transaction given a number of possible transactions however, given real world considerations, sort the possible transactions for a given receipt in order of likelihood of being the correct transaction.
Provide a list of transactions likely to match the receipt, with the one correct at the top of the list. 'Success' in this context means that the correct transaction for the given receipt is at the top of the list

Note: If the correct matching is not in the data for a given receipt 'success' is not possible.

Data Available

The data consists of a csv file with the following fields

Field Name	Description
member_id	tide customer identifier
receipt_id	unique identifier for a receipt image
matched_transaction_id	unique identifier for the transaction, is the correct match for the receipt_id
feature_transaction_id	is the unique_identifier for the transaction which was combined with the receipt_id to produce the matching vector
DateMappingMatch	Likelihood of dates matching
AmountMappingMatch	Likelihood of amount matching
DescriptionMatch	Likelihood of description matching
DifferentPredictedTime	Likelihood of predicted time being different
TimeMappingMatch	Likelihood of time mapping matching
PredictedNameMatch	Likelihood of name matching
ShortNameMatch	Likelihood of short name matching
DifferentPredictedDate	Likelihood of different date being predicted matching
PredictedAmountMatch	Likelihood of different date being predicted matching
PredictedTimeCloseMatch	Likelihood of time being a close match

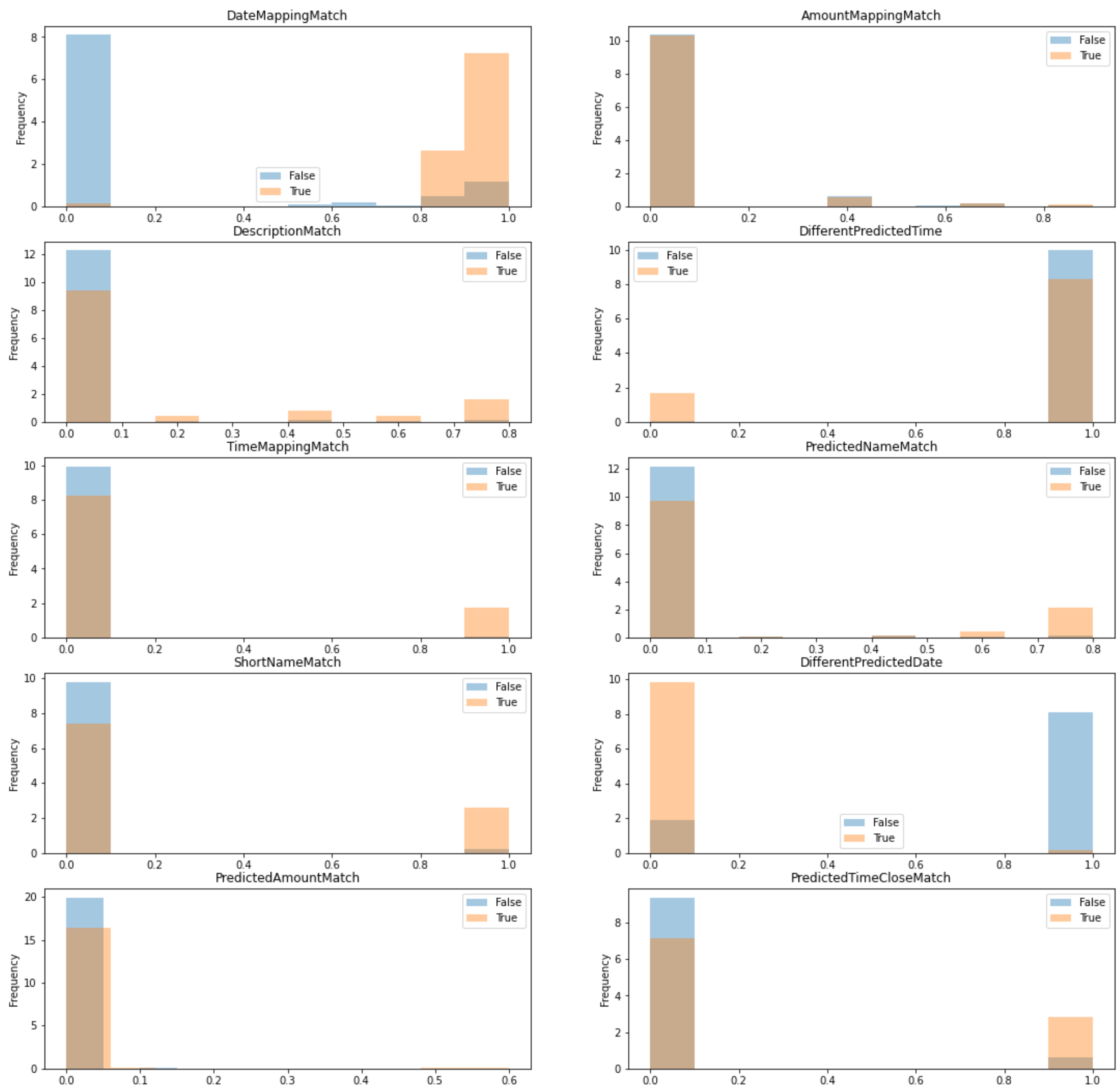
Note : Some filtering was performed in an attempt to reduce the number of receipt-transaction matching vectors therefore not every receipt was matched with every transaction for the member.

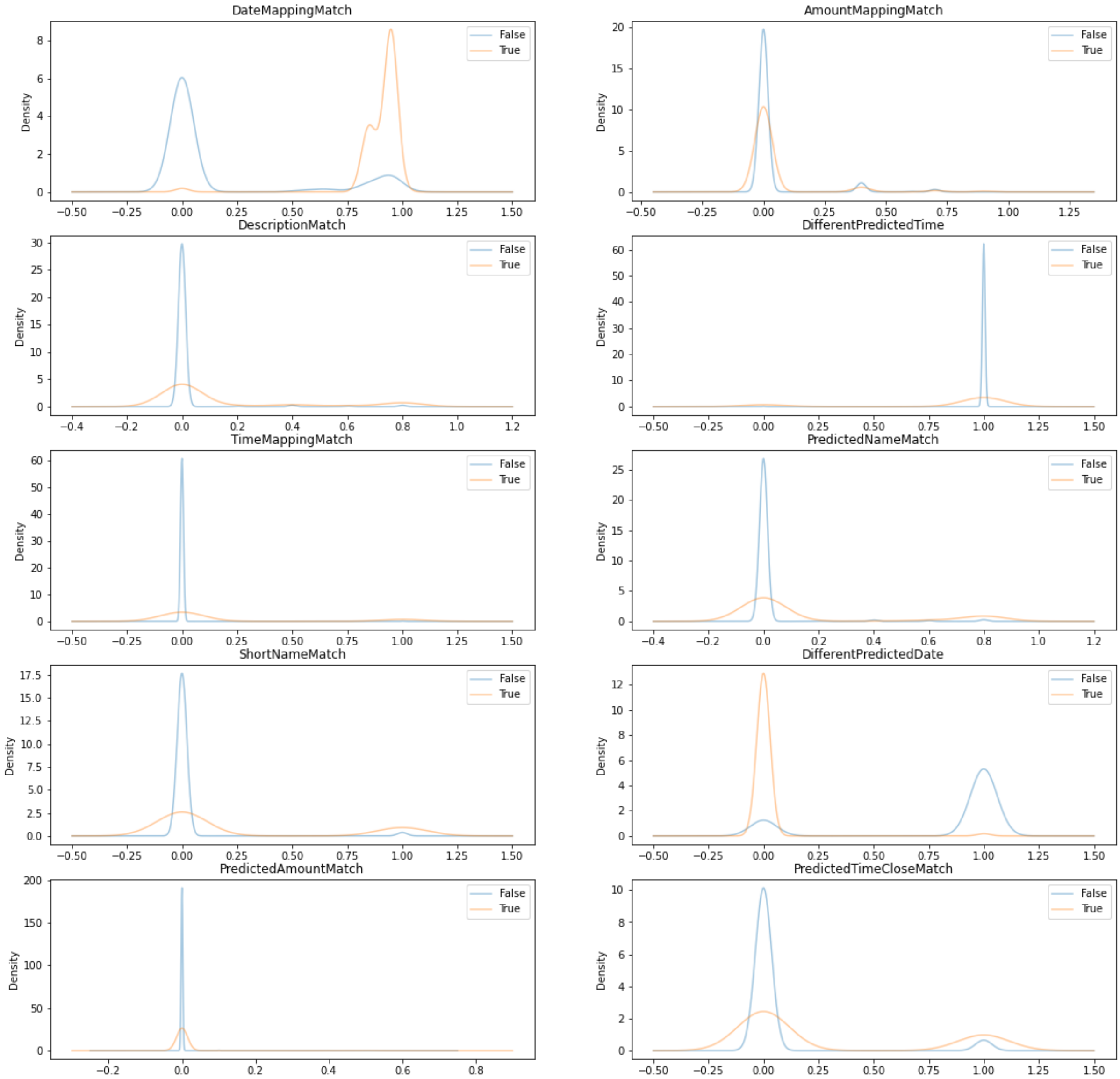
Output definition for Modelling purpose

In order to create a machine learning model, we have to define the output. In the current use case, we will define the output as either 'Success' : denoted as 1 and 'Failure' : denoted as 0. Success is defined when for a row, the matched_transaction_id equals feature_transaction_id, else it is a failure.

Exploratory Data Analysis

We first started off by looking at the distribution of the output variable for each variable available in the data set. Below are histograms and the density plots obtained.





The purpose of the above exercise was to look at the prediction capacity of each variable as well as look at the distribution of the variable. More specifically, wanted to observe if the distribution of the variable was significantly different of the positive and the negative class. The more the difference in the distribution, the better would be the discerning ability of the variable.

Also looking at distribution more quantitatively,

	DateMappingMatch	AmountMappingMa	DescriptionMatch	DifferentPredictedTi	TimeMappingMatch	PredictedNameMa	ShortNameMatch	DifferentPredictedDate	PredictedAmountMatch	PredictedTimeCloseMatch
mean	0.22	0.03	0.02	0.99	0.01	0.02	0.04	0.75	0.00	0.08
std	0.38	0.12	0.12	0.12	0.12	0.13	0.19	0.43	0.02	0.27
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.25	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00
0.50	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00
0.75	0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00
max	1.00	0.90	0.80	1.00	1.00	0.80	1.00	1.00	0.60	1.00

Looking at the above figures, it seems that for some variables the distribution of the success and failure class is different

1. DateMappingMatch
2. DescriptionMatch
3. DifferentPredictedDate
4. DifferentPredictedTime
5. PredictedNameMatch
6. PredictedTimeCloseMatch
7. ShortNameMatch
8. TimeMappingMatch

We could have looked at the difference more quantitatively by looking the distance b/w the distributions, but the purpose of this exercise was to get a rough estimate of the prediction power and sense of the problem.

Also since multiple variables are used to show the closeness of match for date, time and name features, wanted to check if the variables are providing new information or are in principle the same variable, hence next did correlation analysis.

Based on the table given below, we conclude that

1. DifferentPredictedDate
2. DifferentPredictedTime

Do not provide any additional information and can be removed from the data for all future analysis.

	DateMappingMatch	AmountMappingMatch	DescriptionMatch	DifferentPredictedTime	TimeMappingMatch	PredictedNameMatch	ShortNameMatch	DifferentPredictedDate	PredictedAmountMatch	PredictedTimeCloseMatch
company_id	0.06	-0.01	0.05	-0.02	0.02	-0.01	0.04	-0.06	0.01	0.00
DateMappingMatch	1.00	-0.01	0.15	-0.19	0.19	0.18	0.20	-0.99	0.02	0.16
AmountMappingMatch	-0.01	1.00	-0.01	-0.01	0.01	-0.03	0.03	0.01	0.29	0.02
DescriptionMatch	0.15	-0.01	1.00	-0.09	0.09	0.27	0.10	-0.15	0.00	0.08
DifferentPredictedTime	-0.19	-0.01	-0.09	1.00	-0.99	-0.13	-0.16	0.18	0.01	-0.41
TimeMappingMatch	0.19	0.01	0.09	-0.99	1.00	0.13	0.16	-0.18	-0.01	0.40
PredictedNameMatch	0.18	-0.03	0.27	-0.13	0.13	1.00	0.40	-0.17	-0.01	0.08
ShortNameMatch	0.20	0.03	0.10	-0.16	0.16	0.40	1.00	-0.19	0.02	0.07
DifferentPredictedDate	-0.99	0.01	-0.15	0.18	-0.18	-0.17	-0.19	1.00	-0.02	-0.15
PredictedAmountMatch	0.02	0.29	0.00	0.01	-0.01	-0.01	0.02	-0.02	1.00	0.04
PredictedTimeCloseMatch	0.16	0.02	0.08	-0.41	0.40	0.08	0.07	-0.15	0.04	1.00
output	0.50	0.01	0.31	-0.37	0.37	0.30	0.32	-0.48	0.08	0.22

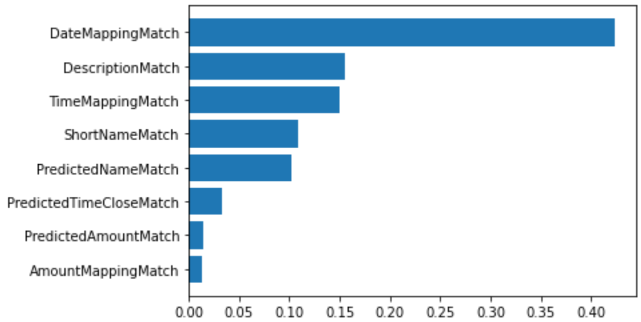
Machine Learning Modelling

Now that we have defined our output and inputs we can proceed to the machine learning modelling part.

Feature selection

To further reduce the feature space, we use feature selection. This step will ensure that the model will generalize better, which means it will perform similarly on future unseen data like it will do during the training phase.
For this purpose used random forest classifier to extract the feature importance of each variable and selected the top 5 variables which could explain 93% of the variance.

```
Explained variance : 0.9398315486068225
Selected features ['PredictedNameMatch' 'ShortNameMatch' 'TimeMappingMatch'
 'DescriptionMatch' 'DateMappingMatch']
```



The above graph shows that DateMappingMatch is the most contributing variable, followed by Description matching. It also makes sense while looking at the feature correlation with the output variable. Given the above analysis we are left with 3 different classes of variables

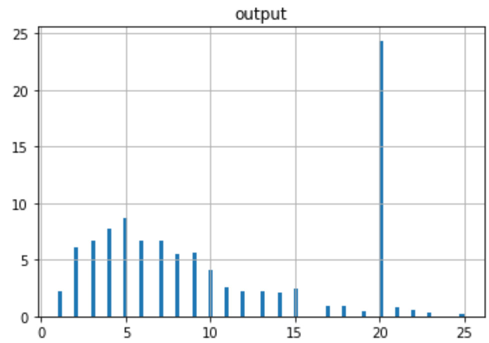
1. Date
    a. DateMappingMatch
2. TextBased
    a. DescriptionMatch
    b. ShortNameMatch
    c. PredictedNameMatch
3. Time
    a. TimeMappingMatch

Also, we have removed redundancy in terms of variables, which gives further confidence that we are heading in the right direction and are not making unnecessary cuts.

Test and Train Split

Our eventual aim is to build a receipt_id level model. Which means we want to give a list of matched transactions for a particular receipt_id in the sorted order. This means that our data should also be split in ratio of receipt ids so that we can look at the final metrics accordingly.

```
Unique receipt_ids in train 924
Unique receipt_ids in test 231
Positive same in train set output    0.070367
Positive same in test set output    0.074722
```



Receipt_ids with more than 3 transactions: 786

The above graphs show the distribution of the transactions matches done for a receipt id.m We observer that for 85% of the receipt ids more than 3 matches were done.
The more the transactions available for a receipt_id the better will be the machine learning model. As already specified in the initial analysis, there is some filtering process done in the way the matching of the transactions was done.
That would lead to inducing bias in the data which can hamper the machine learning technique. However if the prefilter is expected to always induce the same bias, then it won't have a huge negative impact in the overall process apart from constraining the range of the machine learning model.

Modelling process

I experimented with 3 different classification techniques

1. LightGBM
2. NaiveBayes
3. Logistic Regression

The idea was to select a simple model, a Bayesian model and tree based model to compare the accuracy.

While NaivesBayes and Logistic regression don't have many parameters to play around, lightGBM has quite a few parameters to tune. Hence for LightGBM, used Bayesian hyper parameter optimization package called HyperOpt to provide the best hyperparameter values. We could have used also gridsearchCV technique provided natively in the sklearn package.

Results

LightGBM

```
-----Training Result ---
              precision    recall  f1-score   support

       False       0.97      0.99      0.98      9010
        True       0.76      0.58      0.66       682

    accuracy                           0.96      9692
```

```
   macro avg       0.87      0.78      0.82      9692
weighted avg       0.95      0.96      0.96      9692

ROC AUC Score  0.7849076132417223
MCC :  0.6456077181376738

-----Test Result ---
              precision    recall  f1-score   support

       False       0.97      0.98      0.98      2167
        True       0.75      0.63      0.69       175

    accuracy                           0.96      2342
   macro avg       0.86      0.81      0.83      2342
weighted avg       0.95      0.96      0.96      2342

[[2131   36]
 [  65  110]]
MCC :  0.6655457683580995
ROC AUC Score 0.8059792998879293
```

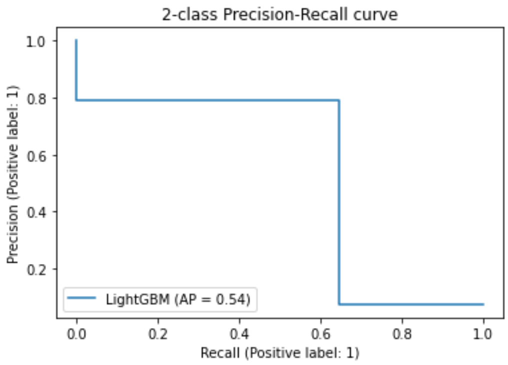
NaiveBayes

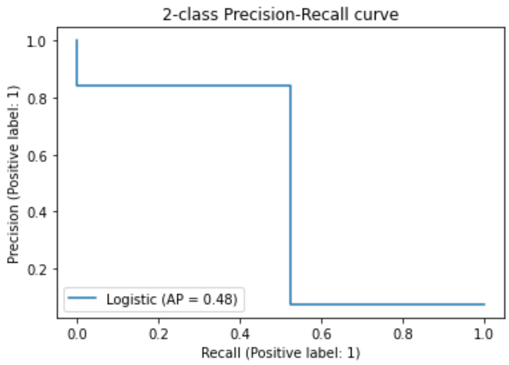
```
precision    recall  f1-score   support

       False       0.97      0.97      0.97      9010
        True       0.63      0.59      0.61       682

    accuracy                           0.95      9692
   macro avg       0.80      0.78      0.79      9692
weighted avg       0.94      0.95      0.95      9692

ROC AUC Score 0.780669734833567
MCC :  0.5777006625340311
-----Test Result ---
              precision    recall  f1-score   support

       False       0.97      0.97      0.97      2167
        True       0.63      0.63      0.63       175

    accuracy                           0.94      2342
   macro avg       0.80      0.80      0.80      2342
weighted avg       0.94      0.94      0.94      2342

MCC :  0.598576043246094
ROC AUC Score 0.799288021623047
```

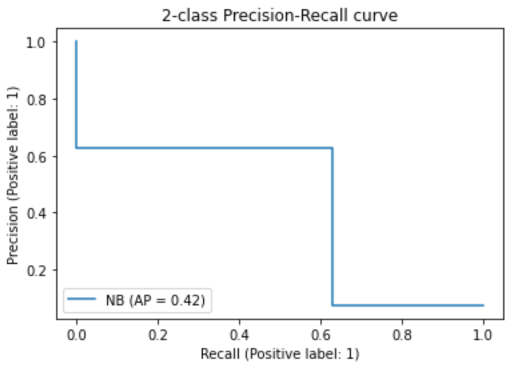
Logistic Regression

```
precision    recall  f1-score   support

       False       0.96      0.99      0.98      9010
        True       0.84      0.48      0.61       682

    accuracy                           0.96      9692
   macro avg       0.90      0.74      0.79      9692
weighted avg       0.95      0.96      0.95      9692

ROC AUC Score  0.7369730927838407
```

```
MCC :  0.6160705010869646
-----Test Result ---
                precision    recall  f1-score   support

       False       0.96      0.99      0.98      2167
        True       0.84      0.53      0.65       175

    accuracy                           0.96      2342
   macro avg       0.90      0.76      0.81      2342
weighted avg       0.95      0.96      0.95      2342

MCC :  0.6464144680147076
ROC AUC Score 0.7589346693915222
```







Given the above results we can conclude that we can pick either of logistic or LightGBM Model since the accuracy is comparable. LightGBM does provide slightly better results which is expected. In case we want our results to be interpretable, we could go with logistic model sacrificing on the accuracy a bit, while if we want to purely focus on accuracy we should select LightGBM model. For future analysis, we select ligthGBM model.

Receipt ID based Analysis

Since our final aim to do receipt_id based analysis, we have to aggregate the results for each receipt_id and check if the top match is indeed the correct match. For this we have to decide on a threshold below which we will say there is no match.
For this analysis, we chose the threshold as 0.2 and 0.1 This can be modified based on the metrics we want to maximize for.

Threshold = 0.2

```
Train receipt_id count 924
Train positive receipt_id 682
Train true positive above threshold 579
Train When  true positive above threshold top rank  394
Train When  true positive above threshold top 2 rank  477
Train false positive above threshold 9

Test receipt_id count 231
Test  positive receipt_id 175
Test  true positive above threshold 153
Test  When  true positive above threshold %age top rank 109
Train When  true positive above threshold top 2 rank  130
Test  false positive above threshold 2
```

Threshold = 0.1

```
Train user count 924
Train positive user 682
Train true positive above threshold 674
Train When  true positive above threshold top rank  409
Train When  true positive above threshold top 2 rank  518
Train false positive above threshold 24

Test user count 231
Test  positive user 175
Test  true positive above threshold 173
Test  When  true positive above threshold top rank 113
Test  When  true positive above threshold top 2 rank 138
Test  false positive above threshold 5
```

Explanation for threshold=0.1:
We have overall 924 receipt_ids in the train sample. Out of which 682 receipt_ids have a Correct transaction_id among the matches done, while the remaining 242 don't have.

For the 682 receipt_ids, in case of 674 receipts_ids the correct transaction is part of the selected transactions and in 409 of these the correct transaction is the top match and 518 of these the correct transaction is in the top 2

In case of the 242 train samples which do not have a correct transaction_id among the matches, 24 of these will be samples will be incorrectly tagged to a transaction id.

So for train sample,
Positive class we can correctly handle 409 out of 682 samples ~ 60% accuracy
Negative class we can correctly handle 218 out of 242 samples ~ 90% accuracy

So for test sample,
Positive class we can correctly handle 113 out of 175 samples ~ 64.5% accuracy
Negative class we can correctly handle 5 out of 56 samples ~ 90% accuracy

Here, by correctly handle we mean, in case of matched transaction, the correct match should be at the top and in case of no match, no transaction should be above the threshold.

The above analysis can be done for various threshold's to determine the sweet spot as per the business use case.

Note : The above accuracy is just of the machine learning model and does not account for Filtering process.

Further improvements

To improve on the above results, the following steps can be undertaken

1. Improve the individual feature accuracy
2. Adding new variables
3. Checking for the bias in initial sample selection