**RECEIPT MATCHING DATA SCIENCE CHALLENGE**

Britto Sabu

horizontal line

# 

# 

# Description

The problem statement is to build a model which can be used to order a set of transactions by likelihood of matching a receipt image feature similarities.

The data provided was a set of transactions and its possible matches and the similarity matrix of those possible matches against the Tide App transaction.

# 

# 

# 

# Approach

**Intuition & Overview**

It starts with the goal of this exercise which is “ Given a set of transaction corresponding to a transaction in the database, find the best possible match and show it on top of the list

Even though the data is at all similar transaction level , the analysis should be done on one matched transaction and the variability of features inside that subset of transactions

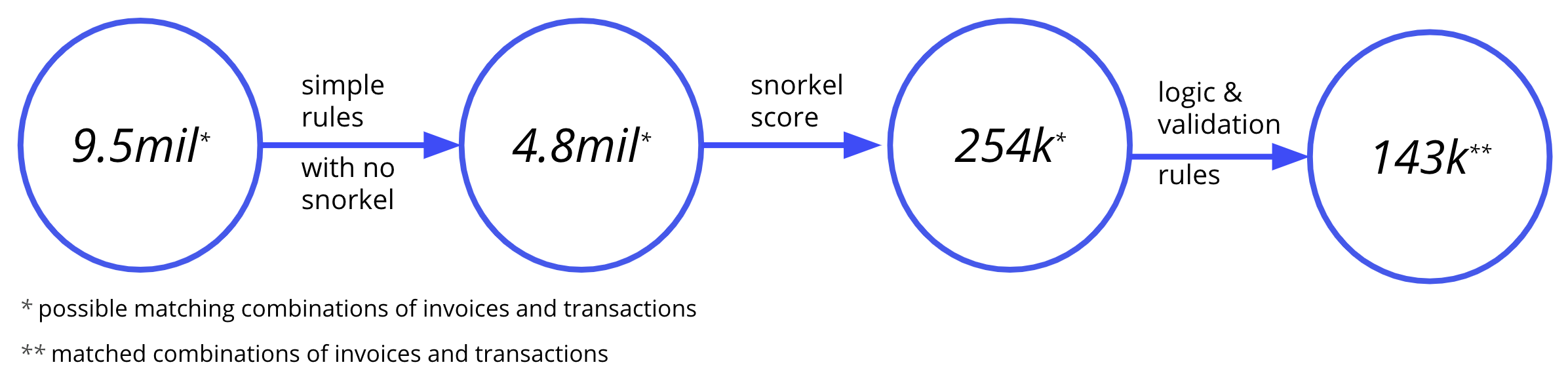
We had 1155 such subsets out of which 298 doesn’t have any matching transaction , 262 had Incorrect matches with the same similarity as Correct matches.

If we train a global model on all transactions without any labels [ or with binary labels] , there can be mismatches since maybe for one subset of transaction almost all the similarity checks were passed but only 1 of them differed which led to the correct match ,

And for another set , only minimal checks were satisfied but among them itself Correct match shined which led to the selection of those . i.e . There can be transactions with similarity match for 7 columns which is a correct transaction and also there can be transactions with similarity match for 2 columns which can be a correct transaction.

So it is always wise to find the value within the subset.

My primary goal was to create a regressor [Keeping this above point in mind]

I’ve also read Tide’s Approach for Reducing the number of invoices for processing using Snorkel [[here](https://medium.com/tide-engineering-team/how-snorkel-a-semi-supervised-learning-technique-solved-invoice-accounting-at-tide-f4a300b8021d)] 

Since the feature columns itself is a Similarity score against the Correct invoice . I’ve taken a **Simple and naive approach** to create a **Similarity Score** for each transaction based on the feature columns which are provided.

Gathering the information which was identified using Data Visualization , i was able to categorize the feature columns into multiple levels [Strong,Normal,Weak] .

I’ve created a **weighted Sum** as a similarity score and checked its efficiency at matched\_transaction\_level .

The **metric** which I **used was Recall** ( Since i believe even if we show false positives on top of the list along with True Positives , it's still better than not being able to show a True Positive on top . So i haven’t cared much about False Negatives as long as more True Positives are coming out on top )

Even with my naive approach (Without optimizing the weights i’ve given [tried using Weight of evidence and Information value but was not working as expected]) , a **Recall score of .91** was obtained.

I’ve used this **score as my benchmark** [ or a regression output] so that i can train an ML model using this output

I’ve used this score to Create a **RandomForestRegressor** , hyperparameter tuning was done and a model was created

The model was tested on top of 100 matched\_transaction\_ids and it gave 87 of them on top of the list which yields **.87 Recall**

# Exploratory Data Analysis

The features[Similarity Matrix] is already a byproduct after matching the tide app transaction with the extracted invoice data. Most of the feature columns are binary or in range(0,1) and no missing values were there . A pretty neat small data

**Observations**

* 298 transactions out of 1155 doesn’t have a matching transaction
* 262 transactions out of the 857 transactions has same similarity feature matrix for correct as well as incorrect transactions
* Correlated Features are present

# Data Preparation

The data itself didn’t explicitly come with a labelled class . But creating labelled class was simple provided we have matching ids in both matching\_transaction\_id and feature\_transaction\_id.

Based on the above observations from EDA , My main focus was on the 595 matched\_transactions in the first place where the feature vector of a correctly matched transaction is not the same as any other transaction in that matched subset.

The feature importances were found using just visualization of the plots based on the labels and used for modelling

Not much Feature engineering techniques were used since the features are already in perfect shape and i didn’t wanted to overcomplicate the modelling process [still i would have done dot product between feature variables to create new features]

# Primary Modelling (Linear)

Using a naive approach of Taking a weighted sum which was solely for creating a regressed score to train ML models .

The model gave a Recall of .91 without much optimization .

Weight of evidence & Information Value didn’t contribute much to improve the model. So had to drop that

# Secondary Modelling

Model which I've used is a RandomForestRegressor . The regressor was able to give .87 recall on the rest data

# Results

The testing of the model was done on 100 randomly selected transactions in which the correct match came up in 87 cases

**Improvements & Suggestions**

* Upon going through the data , I stumbled upon the column ‘PredictedNameMatch’ which I believe is the INVOICE NUMBER matching . In the given sample, it seems like the matched values are not upto mark . Can this be improved by any of these
  + Better images from Tide App which goes into the external supplier [fot invoice to data process]
  + Try to improve the OCR technique used by the external supplier for better results
  + Improved/better Similarity matching for strings