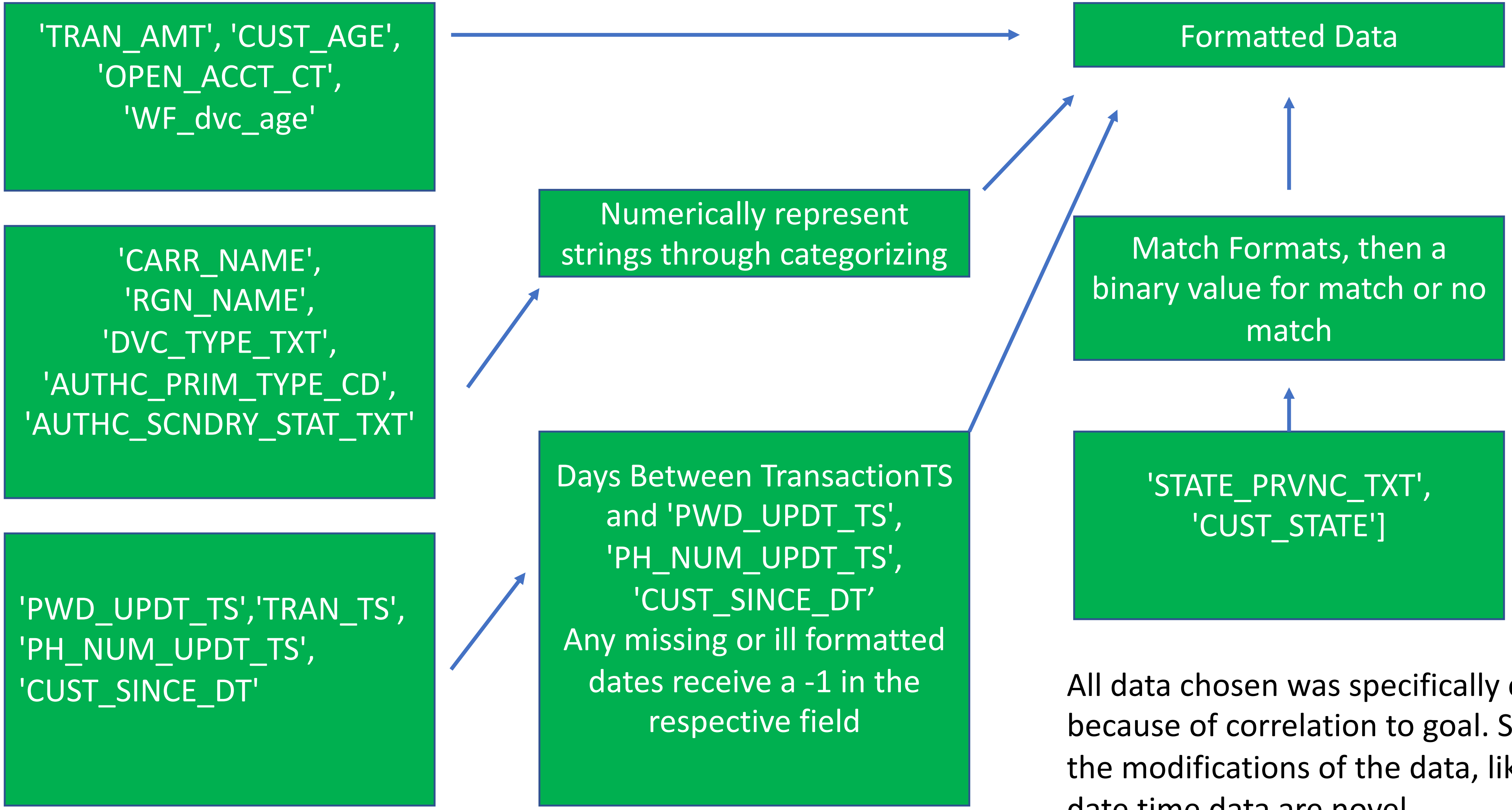
Data Manipulation



'TRAN_AMT', 'CUST_AGE', 'OPEN_ACCT_CT', 'WF_dvc_age'

'CARR_NAME', 'RGN_NAME', 'DVC_TYPE_TXT', 'AUTHC_PRIM_TYPE_CD', 'AUTHC_SCNDRY_STAT_TXT'

'PWD_UPDT_TS','TRAN_TS', 'PH_NUM_UPDT_TS', 'CUST_SINCE_DT'

Numerically represent strings through categorizing

Days Between TransactionTS and 'PWD_UPDT_TS', 'PH_NUM_UPDT_TS', 'CUST_SINCE_DT’ Any missing or ill formatted dates receive a -1 in the respective field

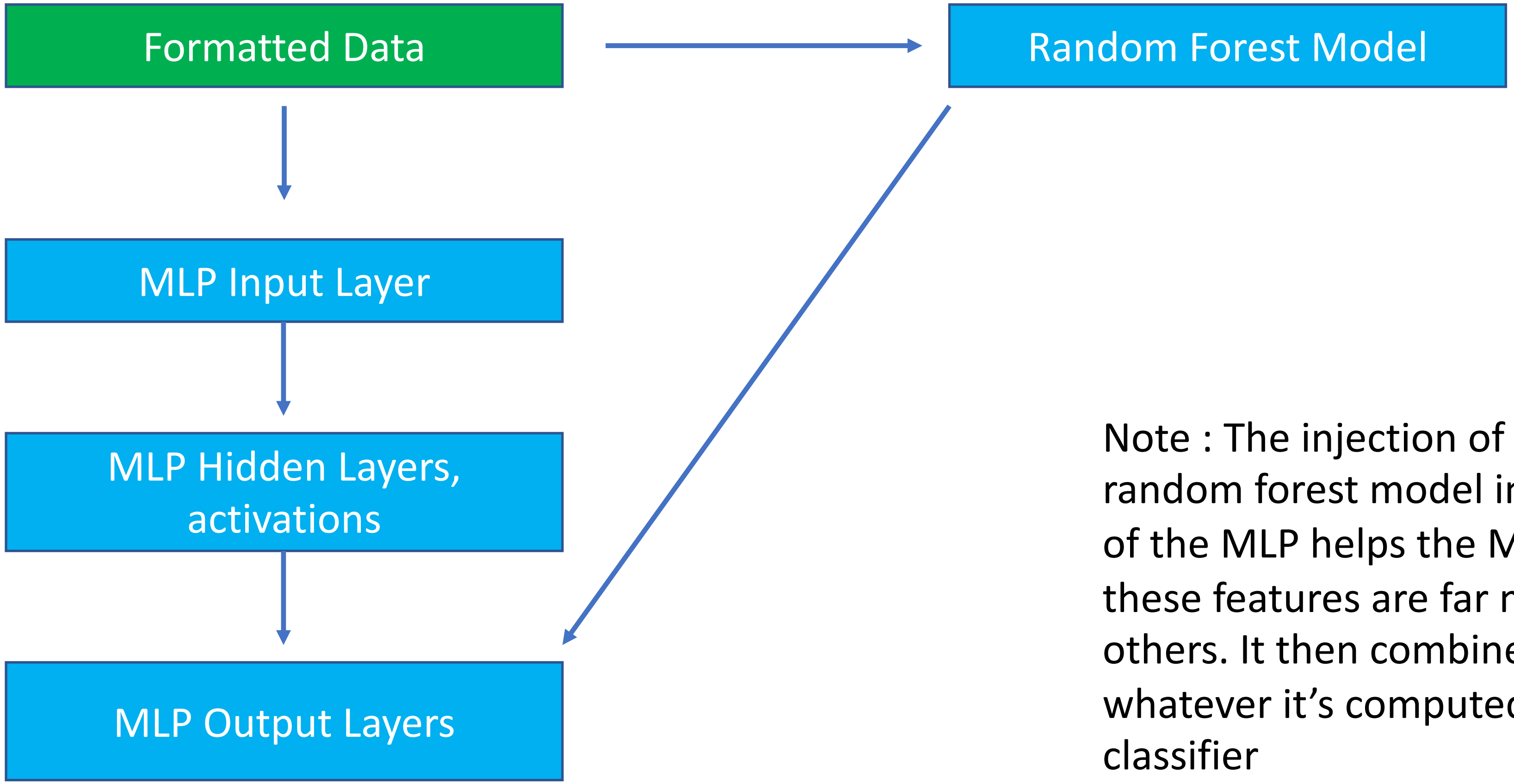
'STATE_PRVNC_TXT', 'CUST_STATE']

Match Formats, then a binary value for match or no match

Formatted Data

All data chosen was specifically chosen because of correlation to goal. Some of the modifications of the data, like the date time data are novel.

MLP Model Structure



Note : The injection of the output of the random forest model into the final stages of the MLP helps the MLP to learn that these features are far more relevant than others. It then combines them with whatever it’s computed for a better classifier

Rationale

When I began the data exploration for this project I started by looking relationships between individual variables and the goal. In some cases there were clear correlations. In many others it appeared as if the variables had very little impact on the outcome. This is where my ideas for data manipulation come from. During this exploration I tried to see what kinds of relationships the variables had to each other, as well as the goal. In particular I examined the date and time data. Rather than examining these data point on their own I tried to see if the number of days between different fields had a stronger correlation. In nearly every case it did. The final adjustment I made to the data was to examine the state of the customer and the state of the transaction. Alone, each did nothing. But, by comparing them to each other I was given a very strong correlation with the outcome. This was how I went about creating the input data.

During my model exploration I tried many simple models to test the data. By far, out of the simple models, the random forest model took the cake. The other models had an F1 score around .88 while the random forest had one around .95. Obviously, the random forest had to be the center of my predictive modeling scheme. When I thought about how to combine these models, it dawned on me that it wouldn't be effective to merely average the predictions. The random forest was too much better than the others, and the accuracy of the averaged model was always below that of the random forest. This led me to the idea of using it as an input in an MLP. Initially, I just added the random forest output to the input layer. This proved to be very ineffective. The model stayed at around an F1 score of .86. I suspect that it was having trouble seeing just how much of a relationship these values held to the output. To correct for this, I decided to attempt inserting it later into the model where it's effects would be felt stronger. This is how I came up with the structure of the model I submitted.

The strength of this method is that it allows an MLP model to build on the work that another model has done. The MLP performs most of the calculations itself, and then uses the output of the random forest in the final stage. This works as a method of confirmation. If both agree then the odds that the model is correct is substantially higher. It also provides the opportunity to analyze the weaknesses of the random forest model, and try to correct for them. That is why the f1 score for this was around .975. The weakness of this type of model is that it depends much more heavily on the previous model. If the initial random forest did not perform well, then the model built on top of it would not succeed either.

Because the F1 score is high, this has the potential to be used as a real world predictor of elder fraud. This, combined with a different model with a similar F1 score could allow for a very high accuracy.