Ziwan Peng
pengz21@wfu.edu
10/24/2021

Executive Summary

Problem

As a data scientist in a financial institution, I am hired to develop a model to identify transactions that are likely to be fraud because fraud transactions will impact heavily on the company's total revenue. False prediction on actual legit transactions will cause much more loss than false prediction on actual fraud transactions because financial institution would refund to card holders when the transaction is reported as fraud. More false fraud detections will lead to more loss for the financial institution. Therefore, we need to consider more about how to decrease false prediction on actual legit transactions when developing model. With the final model, we have a clearer idea of prediction on fraud.

Key Findings

- Email domains matter in prediction on fraud transactions. Transactions with email domains such as larson-harris.com, brady.org and davies.org are 10 times more likely to be fraud than transactions with other email domains.
- Billing postal code can be considered as an important predictor because there are some postal codes have more than 60% probability to generate fraud transactions.
- IP addresses of some transactors can approach to 80% probability to report fraud transactions.
- The probability of fraud with USD currency is approximately 1 times higher than that with Euro currency.

Model Performance Summary & Interpretation

Because our data set is heavily imbalanced, with 5.43% fraud and 94.57% legit, we consider the AUC score instead of accuracy as key metrics to measure model performance. The final lasso logistic regression model has a great performance with extremely high AUC score 94.8% in test data set. It means our model can achieve an extremely high classification accuracy in predicting fraud and legit.

In this case, because the financial institution will refund to card holders when the transaction is reported as fraud, the financial institution would lose revenue on fraud transactions and benefit more on legit transactions. False positive means our model misclassifies actual legit transactions as fraud, which leads the financial institution refunds the money that shouldn't be refunded or cancels the transaction that shouldn't be canceled so that it causes lots of loss on revenue. And false negatives don't lead to such serious consequences. Therefore, we should consider more on false positive than false negative. The company aims a 6% false positive rate, meaning that the company only allows maximum 6% of legit to be misclassified as fraud, to limit the loss due to false fraud detections.

Under this consideration, we change the threshold to 0.395. It means if the score (probability of fraud in this transaction) is higher than the threshold, the transaction would be reported as fraud. Otherwise, it will be detected as legit.

Recommendations

- The company can create some lists to record some suspicious email domains, billing postal codes and IP addresses because they all have more than 10 times higher fraud rates. Once there's a transaction with email domain or IP address in those suspicious lists, the company should reject or

set constraints on the transaction until further evidence is provided like double check with phone calls.
- The company can provide more bonus like higher cash back for transactors who use USD dollars and more restrictions like only Mastercard or higher commission fee on Euro.

Model Comparison

I employed four models: lasso logistic regression, ridge logistic regression, decision tree and random forest models following the same model training steps as above. Here is their performance metrics. Because in this case false positive matters most and is required to achieve 6%, I would consider it more when choosing the optimal threshold.
According the comparison table, we can observe that:

a. Lasso regression performs best in both train data and test data and it's easier to interpret and more friendly to business executives with limited knowledge in machine learning.
b. Ridge regression has great performances on precision rate but lower scores on AUC and recall rate than lasso regression.
c. Decision tree model has a better performance on recall and precision rate with same false positive rate but lowest AUC scores on train and test data set.
d. Random Forest has the best performance in train data set but larger cap in AUC score between train data and test data. With the same 6% false positive rate, it has the lowest performance on recall and precision rate.

Therefore, I picked lasso logistic regression as the final model because the performance on the test data matters most.

Model	**Train AUC**	**Test AUC**	**Recall**	**Precision**	**threshold**	**False positive rate**
lasso regression	0.94300158	0.94798967	0.66	0.853	0.395	0.06
ridge regression	0.942775	0.94735	0.64	0.857	0.343	0.06
decision tree	0.92079964	0.9260073	0.665	0.856	0.352	0.06
random forest	0.95400884	0.94028196	0.62	0.843	0.22	0.06

Detailed Analysis & Steps

File Summary

File Name	Record count	Column count	Numeric columns	Character columns	Timestamp columns
project_2_training.csv	125000	27	9	17	1

Field Summary

Character variables

	Variable Name	Data Type	Feature Type	Count	Unique	N_Null	Missing Rate
1	ip_address	char	categorical	125000	13314	0	0.0000%
2	user_agent	char	categorical	125000	8571	0	0.0000%
3	email_domain	char	categorical	125000	6992	0	0.0000%
4	phone_number	char	categorical	125000	11928	0	0.0000%
5	billing_city	char	categorical	125000	8980	0	0.0000%
6	billing_state	char	categorical	125000	51	0	0.0000%

7	currency	char	categorical	125000	4	0	0.0000%
8	cvv	char	categorical	125000	26	0	0.0000%
9	signature_image	char	categorical	125000	27	0	0.0000%
10	transaction_type	char	categorical	125000	27	0	0.0000%
11	transaction_env	char	categorical	125000	27	0	0.0000%
12	applicant_name	char	text	124876	84958	124	0.0992%
13	billing_address	char	text	124889	124884	111	0.0888%
14	merchant_id	char	id	124911	124904	89	0.0712%
15	locale	char	categorical	124885	293	115	0.0920%
16	tranaction_initiate	char	categorical	124900	26	100	0.0800%
17	event_label	char	categorical, target	125000	2	0	0.0000%

Timestamp variable

Variable Name	N_null	Missing Rate	Mix	Max	Median	N_Unique
event_timestamp	90	0.0720%	10/25/20 8:44:38	10/25/21 14:27:09	4/25/21 23:50:23	124685

Numeric variables

	Variable Name	Data Type	Feature Type	N null	Missing Rate	Mean	STD	Min	Median	Max
1	event_id	numeric	id	0	0.0000%	1500444	866357	20	1500570	2999960
2	account_age_days	numeric	numeric	0	0.0000%	4642	1161	-1	4668	9119
3	transaction_amt	numeric	numeric	0	0.0000%	2520	609	-1	2543	4880
4	transaction_adj_amt	numeric	numeric	0	0.0000%	54.1	10.2	-1	55	99
5	historic_velocity	numeric	numeric	0	0.0000%	4700	1194	-1	4731	8875
6	billing_postal	numeric	categorical	98	0.0784%	50211	28406	503	50124	99950
7	card_bin	numeric	categorical	110	0.0880%	41813	10084	6040	42061	67639
8	days_since_last_logon	numeric	numeric	113	0.0904%	49.8	29.2	0	50	100
9	inital_amount	numeric	numeric	109	0.0872%	8000	4050	1000	8007	15000

Target Summary

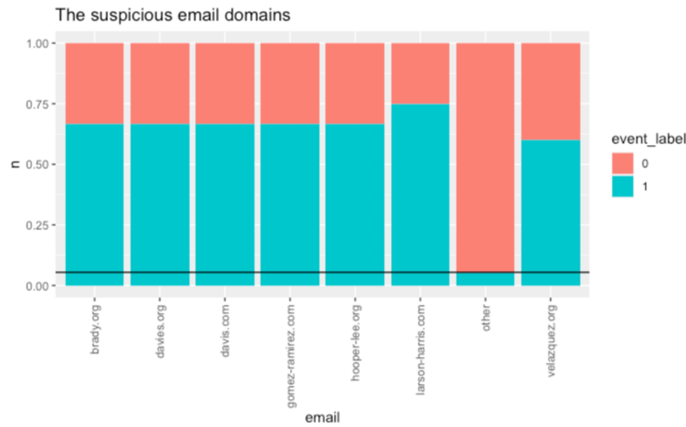
In the training date set, fraud transactions take up only 5.43% and legit transactions take up 94.57%. The large cap between target variable drives us to pay attention to potential issues that may be caused by the imbalance.

fraud	n	Percentage
0	118215	94.57%
1	6785	5.43%

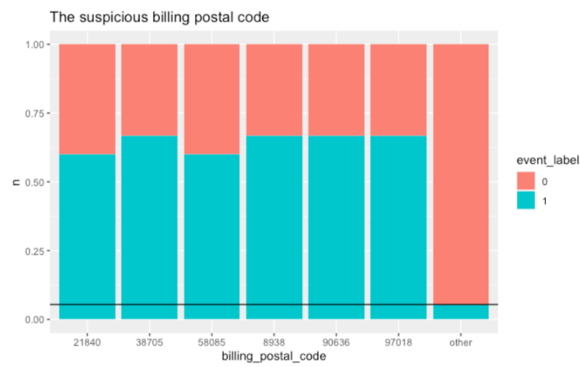
Exploratory Data Analysis & Screening

Categorical variables

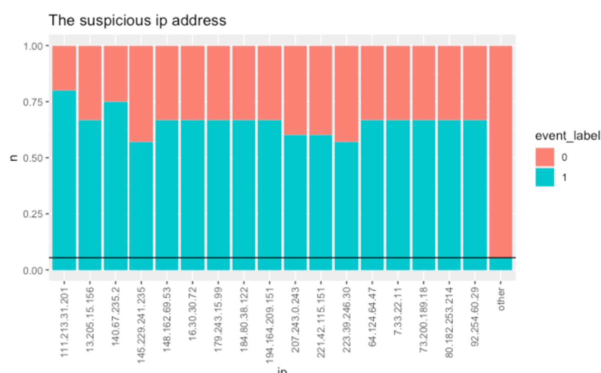
a. Email domain matters in fraud detections. 7 email domains have more than 50% fraud rate, 10 times higher than the average fraud rate 5%.



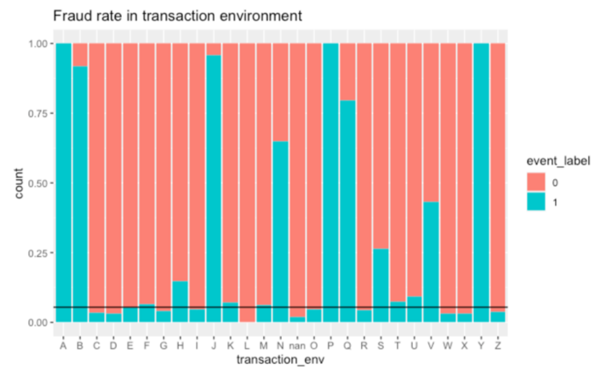
b. Billing postal code can be considered as an important predictor in fraud detections because there's 6 suspicious billing postal codes with >50% probability to report fraud transactions.



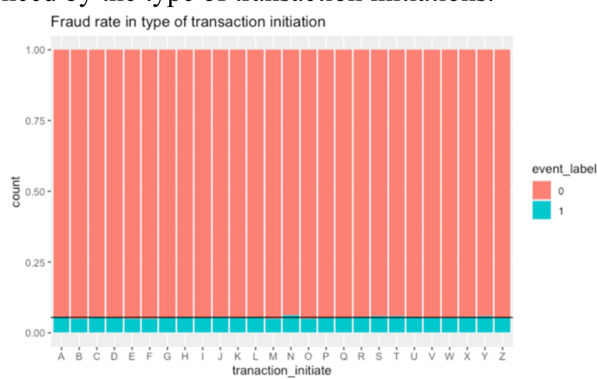
c. IP address of transactors in the following list are in much higher risk of fraud transactions than other IP addresses.



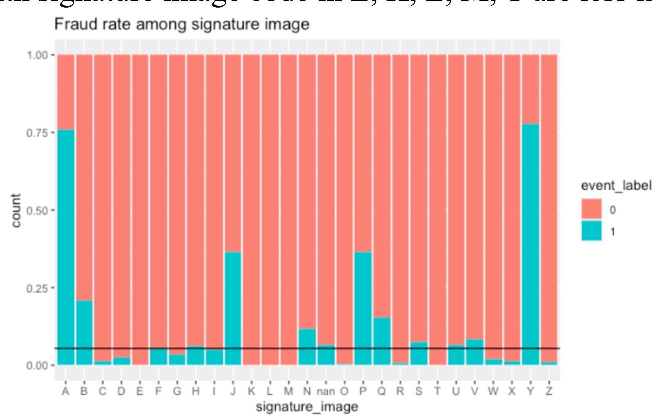
d. Transaction in environment code A, B, J, N, P, Q, Y are much more likely to be fraud, especially for environment code Q, P, Y with fraud rate 100%.



e. Fraud is not influenced by the type of transaction initiations.

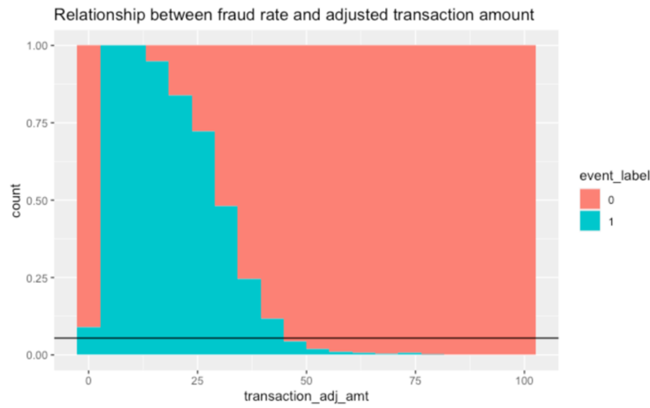


f. Transactions with signature image code in E, K, L, M, T are less likely to be fraud.



Numerical variables

a. There is a clear negative relationship between fraud rate and adjustment amount to the transaction. As adjustment amount increases, the probability to be fraud decreases. The slope is steeper in smaller adjustment amount.



b. There's a positive relationship between fraud rate and number of days since the account was created. With longer time since the account was created, transactions are more likely to be fraud.

c. No significant difference in fraud rates among the amount of first transaction. We can remove this variable in the following modeling.

Data Preparation & Transformation

Variable selection

a. I set event_id as id variables, not predictors in model.
b. I removed user_agent, phone_number, billing_city, card_bin, applicant_name, billing_address, merchant_id, locale, tranaction_initiate, days_since_last_logon, inital_amount because they are not useful factors.

Transformation

a. Factor: Transform all categorical variables into dummy variables

b. Normalize: Because numeric variables have large different scales, we normalize them in case some variables will overweight in our model.

Missing Values

a. For categorical variables, I impute mode to replace null values.
b. For numeric variables, I impute median to replace nulls to avoid influence by extreme values.

Derive new variables

I extract the year, quarter, month, day, weekdays, hour out of event_timestamp as categorical variables year, quarter, month, day, weekdays, hour. But from graphs, we don't need to consider any because they don't have significant impacts on fraud rate.

Model Building

Spliting dataset into train and test set as 80:20.

Model Training

Variables used in modeling

	Variable Name	Data Type	Feature Type	Status
1	ip_address	char	categorical	Relabeled as ip
2	user_agent	char	categorical	rejected
3	email_domain	char	categorical	Relabeled as email
4	phone_number	char	categorical	rejected
5	billing_city	char	categorical	rejected
6	billing_state	char	categorical	
7	currency	char	categorical	
8	cvv	char	categorical	
9	signature_image	char	categorical	
10	transaction_type	char	categorical	
11	transaction_env	char	categorical	
12	applicant_name	char	text	rejected
13	billing_address	char	text	rejected
14	merchant_id	char	id	rejected
15	locale	char	categorical	rejected
16	tranaction_initiate	char	categorical	rejected
17	event_label	char	categorical	target
18	event_id	numeric	id	id var
19	account_age_days	numeric	numeric	
20	transaction_amt	numeric	numeric	
21	transaction_adj_amt	numeric	numeric	
22	historic_velocity	numeric	numeric	
23	billing_postal	numeric	categorical	Relabeled as billing_postal_code
24	card_bin	numeric	categorical	rejected
25	days_since_last_logon	numeric	numeric	rejected
26	inital_amount	numeric	numeric	rejected
27	year	numeric	categorical	rejected
28	quarter	numeric	categorical	rejected
29	month	numeric	categorical	rejected
30	day	numeric	categorical	rejected
31	weekday	numeric	categorical	rejected
32	hour	numeric	categorical	rejected
33	event_timestamp	timestamp	timestamp	rejected

Define your Recipe

```r
fraud_recipe <- recipe(event_label ~ .,
                       data = train) %>%
  step_rm(ip_address,user_agent,email_domain,phone_number,billing_city,billing_postal,
          card_bin,applicant_name,billing_address,merchant_id,locale,tranaction_initiate,
          year,quarter,month,day,weekday,hour,event_timestamp,days_since_last_logon,inital_amount)%>%
  update_role(event_id, new_role = "id variable")%>%
  step_novel(all_nominal_predictors()) %>%
  step_impute_median(all_numeric_predictors()) %>%
  step_impute_mode(all_nominal_predictors()) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_dummy(all_nominal_predictors())
```

Note: I removed the original ip_address, email_domain, billing_postal columns in recipe because I created three new columns named ip, email, billing_postal_code based on suspicious lists and relabeled those suspicious levels. So the final model still includes those variables just not with original categories.

Define your Model

a. Create a workflow and Fit the model

```r
lg1<- logistic_reg(penalty = 0.001, mixture = 1) %>%
  set_mode("classification") %>%
  set_engine("glmnet")

logistic_wf <- workflow() %>%
  add_recipe(fraud_recipe) %>%
  add_model(lg1) %>%
  fit(train)
```

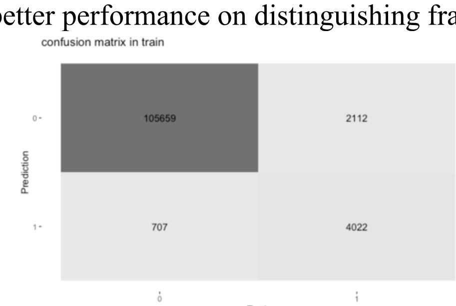
b. Evaluate metrics on Train and Test:
From the metrics table, the lasso logistic regression model has a very great performance in test data set with 97.41% accuracy rate and 94.8% AUC score.

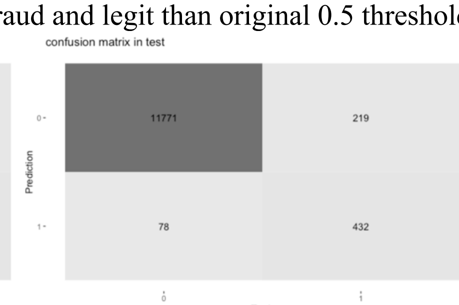
Lasso Logistic Regression	Train	Test
AUC	0.94300158	0.94798967
Accuracy	0.97367111	0.97408000

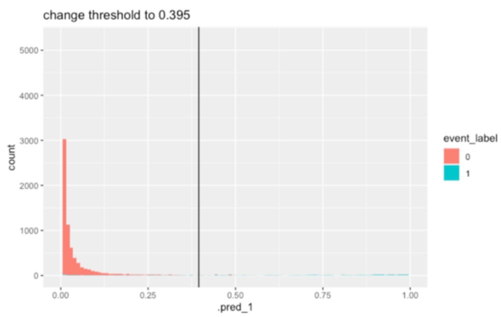
Because the imbalanced data set, we change the threshold to minimize our loss caused by false positive with formula false_positive_rate=round(6134*recall*(1/precision-1)/11849,2). As we can see from the score distribution graph, threshold with 0.395 generates 6% false positive rate.

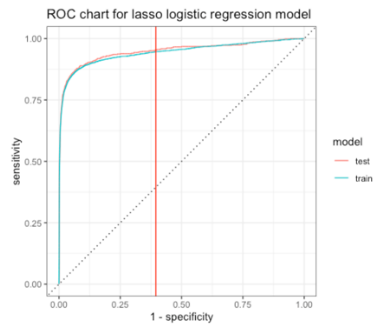
recall	precision	.threshold	false_positive_rate
0.66	0.853	0.395	0.06
0.68	0.837	0.357	0.07
0.69	0.835	0.344	0.07
0.7	0.831	0.33	0.07
0.71	0.82	0.316	0.08

The following ROC chart, confusion matrix and distribution graph with updated threshold 0.395 reached better performance on distinguishing fraud and legit than original 0.5 threshold.

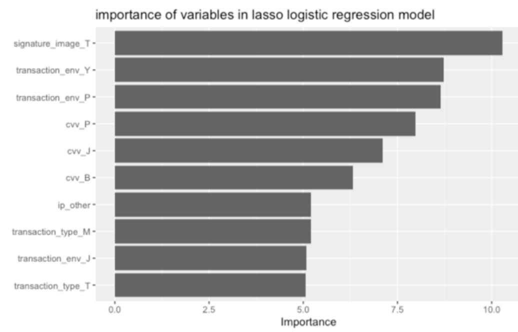








From bar chart of the importance of variables, we can see categorical variables such as signature image with code T, transaction environment Y, P do play important roles on the fraud rate, which confirms the observations in the exploratory data analysis.



From the terms table, we can see the IP address X73.200.189.18, email domain gomez.ramire.com and billing postal code X58085 all have significantly positive relationships with fraud rate. As a transaction with the above information, the fraud rate increases.

term <chr>	estimate <dbl>
transaction_env_N	2.771
cvv_V	2.427
ip_X73.200.189.18	2.383
transaction_type_P	2.320
email_gomez.ramirez.com	2.184
billing_postal_code_X58085	2.178
cvv_S	2.137
transaction_type_B	2.035
transaction_env_V	1.983
signature_image_J	1.924

Model Comparison

I employed four models: lasso logistic regression, ridge logistic regression, decision tree and random forest models following the same model training steps as above. Here is their performance metrics. Because in this case false positive matters most and is required to achieve 6%, I would consider it more when choosing the optimal threshold.

According the comparison table, we can observe that:

e. Lasso regression performs best in both train data and test data and it’s easier to interpret and more friendly to business executives with limited knowledge in machine learning.
f. Ridge regression has great performances on precision rate but lower scores on AUC and recall rate than lasso regression.
g. Decision tree model has a better performance on recall and precision rate with same false positive rate but lowest AUC scores on train and test data set.
h. Random Forest has the best performance in train data set but larger cap in AUC score between train data and test data. With the same 6% false positive rate, it has the lowest performance on recall and precision rate.

Therefore, I picked lasso logistic regression as the final model because the performance on the test data matters most..

Model	Train AUC	Test AUC	Recall	Precision	threshold	False positive rate
lasso regression	0.94300158	0.94798967	0.66	0.853	0.395	0.06
ridge regression	0.942775	0.94735	0.64	0.857	0.343	0.06
decision tree	0.92079964	0.9260073	0.665	0.856	0.352	0.06
random forest	0.95400884	0.94028196	0.62	0.843	0.22	0.06