Credit Card Fraud Can Be **Fun!**

Brian Kim
6/18/2018

Inspiration

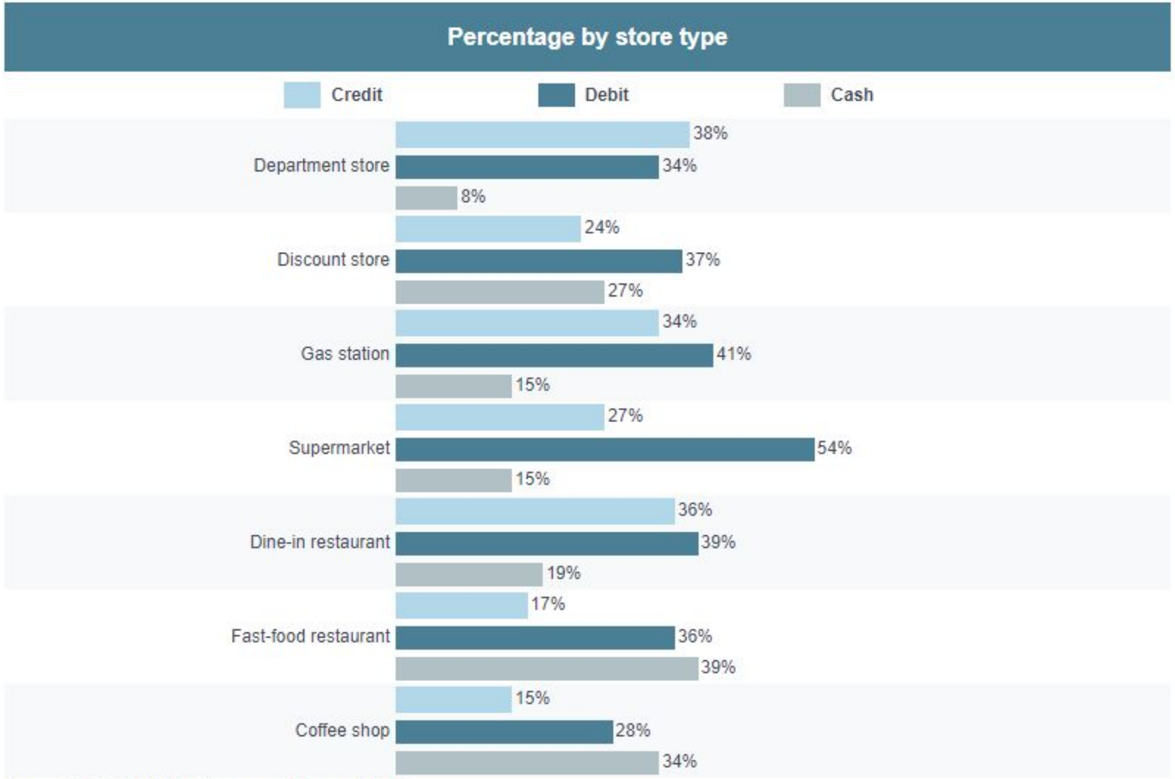
- Family & Community
- Access to Money
- Money
- Money
- Money

Thinkful’s Potpourri

- Credit Card Fraud

Why Are Credit Cards Important?

- To rack up debt to buy things we can't afford
- Opportunity cost
- Credit Cards are an important part of our lives in the US
- Potential locations where Fraud can occur



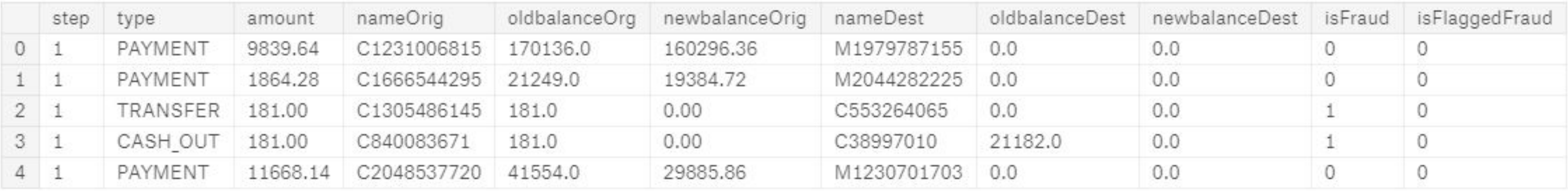
Source: TSYS 2017 U.S. Consumer Payment Study

Can I Create a Model to Predict Credit Card Fraud?

Where’s the data

- Search on Google
    - Credit card transaction data
        - People asking each other where to get data
        - Data.gov
- Kaggle
    - Can’t use Credit Card Fraud Detection
        - Part of the potpourri
    - Synthetic data from financial payment system
        - https://www.kaggle.com/ntnu-testimon/paysim1

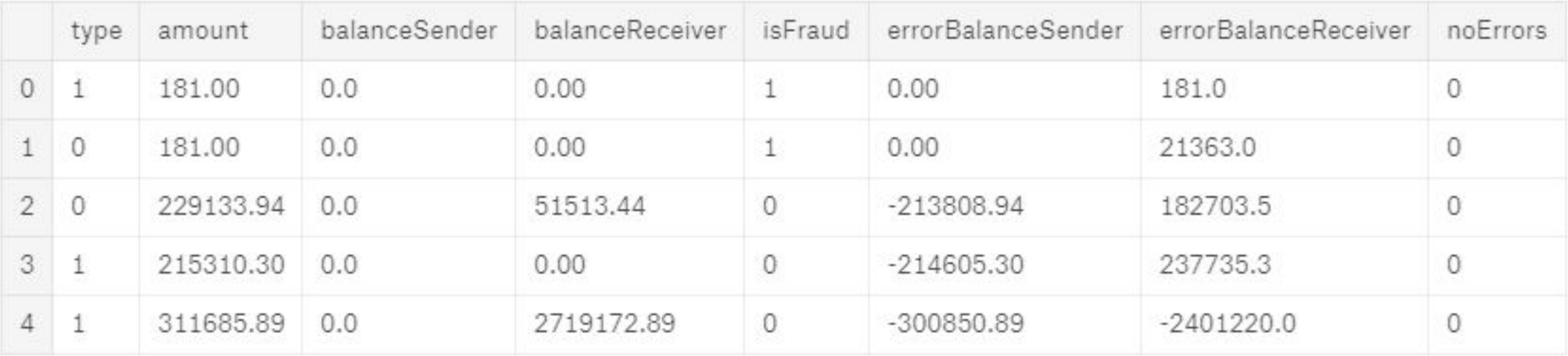
Data Review



	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

- **step** (int): Unit of time in real world. 1 step = 1 hour of time. Total steps is 744 (31 days of simulation)
- **type** (object): CASH-IN, CASH-OUT, DEBIT, PAYMENT, TRANSFER
- **amount** (amount): amount of transaction in local currency
- **nameOrig** (object): customer who started transaction
- **oldBalanceOrg** (float): initial balance before transaction
- **newBalanceOrig** (float): new balance before transaction
- **nameDest** (object): customer who receives transaction
- **oldBalanceDest** (float): initial balance before transaction. No information for customers who starts with M (merchants)
- **newBalanceDest** (float): new balance before transaction. No information for customers who starts with M (merchants)
- **isFraud** (int): marks whether transactions are fraud
- **isFlaggedFraud** (int): marks whether a transaction is an illegal attempt. An illegal attempt is attempting to transfer more than 200,000 in a single transaction

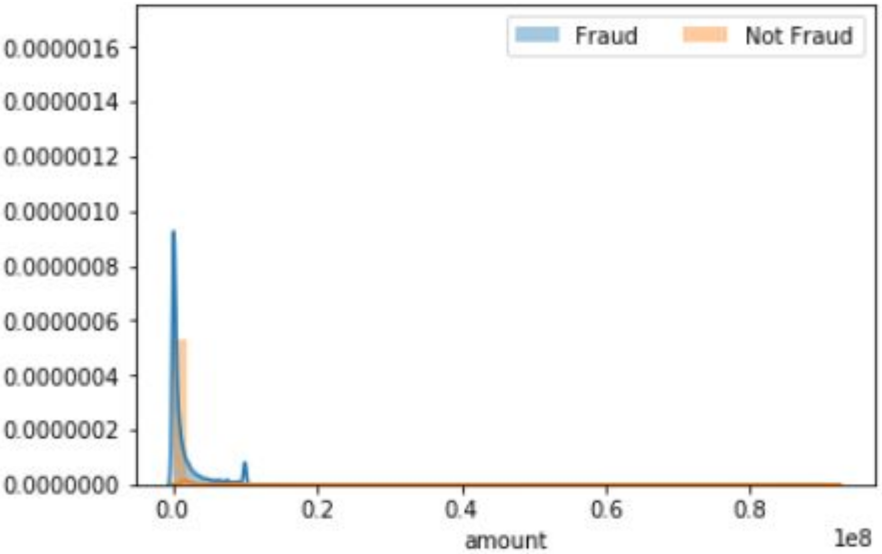
Data Cleaning

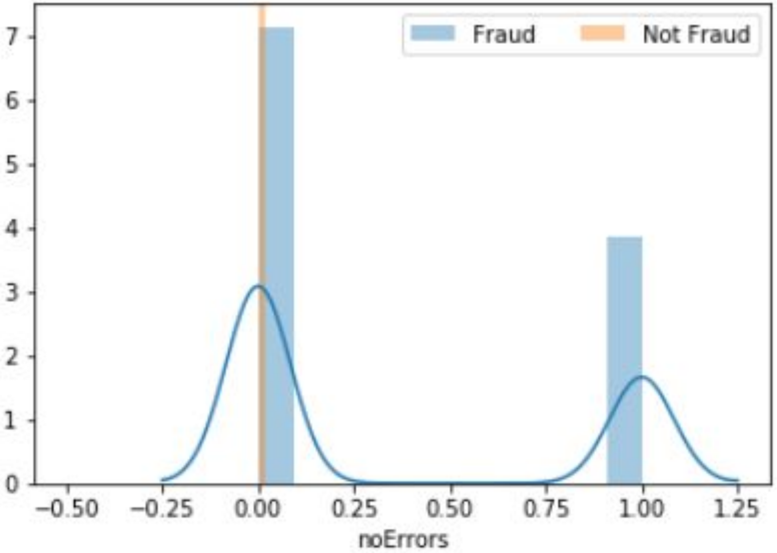


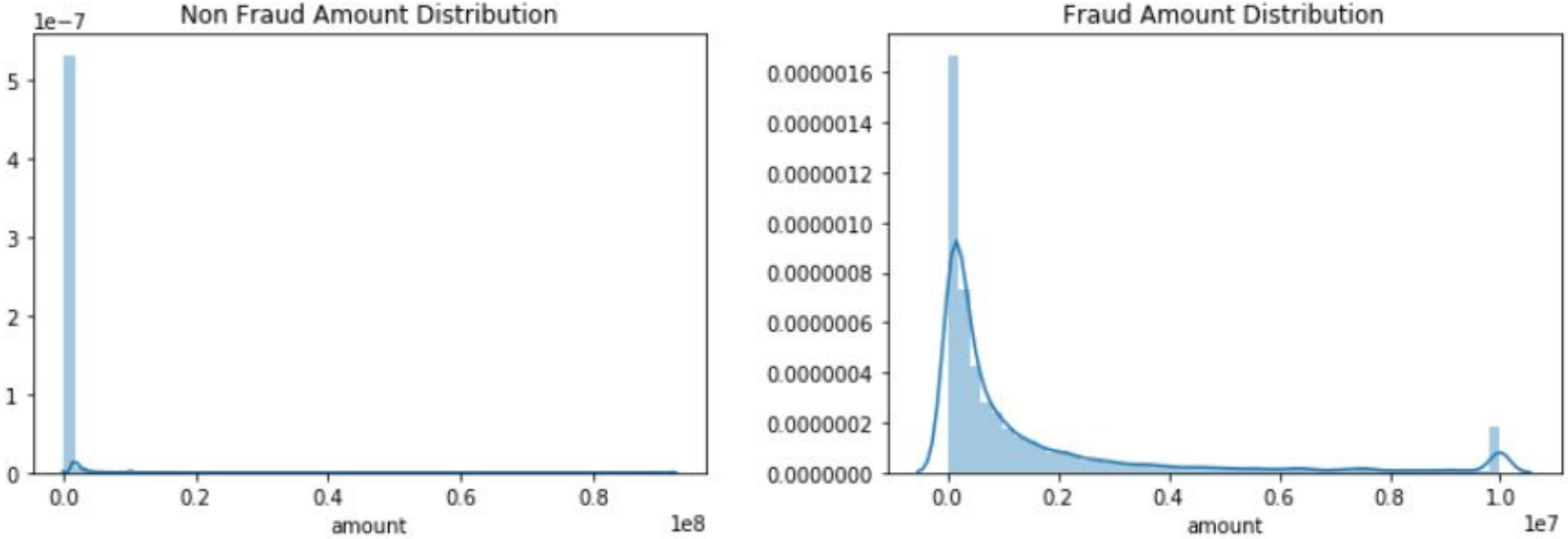
	type	amount	balanceSender	balanceReceiver	isFraud	errorBalanceSender	errorBalanceReceiver	noErrors
0	1	181.00	0.0	0.00	1	0.00	181.0	0
1	0	181.00	0.0	0.00	1	0.00	21363.0	0
2	0	229133.94	0.0	51513.44	0	-213808.94	182703.5	0
3	1	215310.30	0.0	0.00	0	-214605.30	237735.3	0
4	1	311685.89	0.0	2719172.89	0	-300850.89	-2401220.0	0

- **type** (int): CASH-OUT (0), TRANSFER (1)
- **amount** (float): amount of transaction in local currency
- **balanceSender** (float): ending balance of sender
- **balanceReceiver** (float): ending balance of receiver
- **isFraud** (int): marks whether transactions are fraud
- **errorBalanceSender** (float): discrepancy between previous oldBalanceOrg and newBalanceOrig
- **errorBalanceReceiver** (float): discrepancy between previous oldBalanceDest and newBalanceDest
- **noErrors** (int): Checks if errorBalanceSender and errorBalanceReceiver are 0

Distributions






Non Fraud Amount Distribution
1e−7
5
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
amount
1e8
Fraud Amount Distribution
0.0000016
0.0000014
0.0000012
0.0000010
0.0000008
0.0000006
0.0000004
0.0000002
0.0000000
0.0
0.2
0.4
0.6
0.8
1.0
amount
1e7

To Predict Credit Card Fraud...

- Logistic Regression Model
  - Tends to work great with Binary Classifications
- Random Forest Model
  - Good at finding probability of belonging to a classification
- Gradient Boosting Model
  - Great with finding which features are most important

- Different parameters for models
- Different train/test size
- Different sample size
- Reducing Features

Logistic Regression

- **Inverse of regularization strength**
- C= 1e10
- 1e5
- 1 (default)
- 1e-5
- 1e-10

- **Penalty Type**
- L1
- L2 (default)

- **Max iteration**
- 10
- 100 (default)
- 200

Default parameters are equivalent or better than alternative parameters

Random Forest

- **Max features**
- Auto (default)
- None
- Half

- **Max depth**
- 2
- 4
- 6
- 8
- None (default)

- **Number of trees**
- 5
- 10 (default)
- 15
- 20

Max depth of 8 is only parameter that performs better than default parameters

Gradient Boosting

- **Learning rate**
- 0.1 (default)
- 0.25
- 0.5
- 0.75

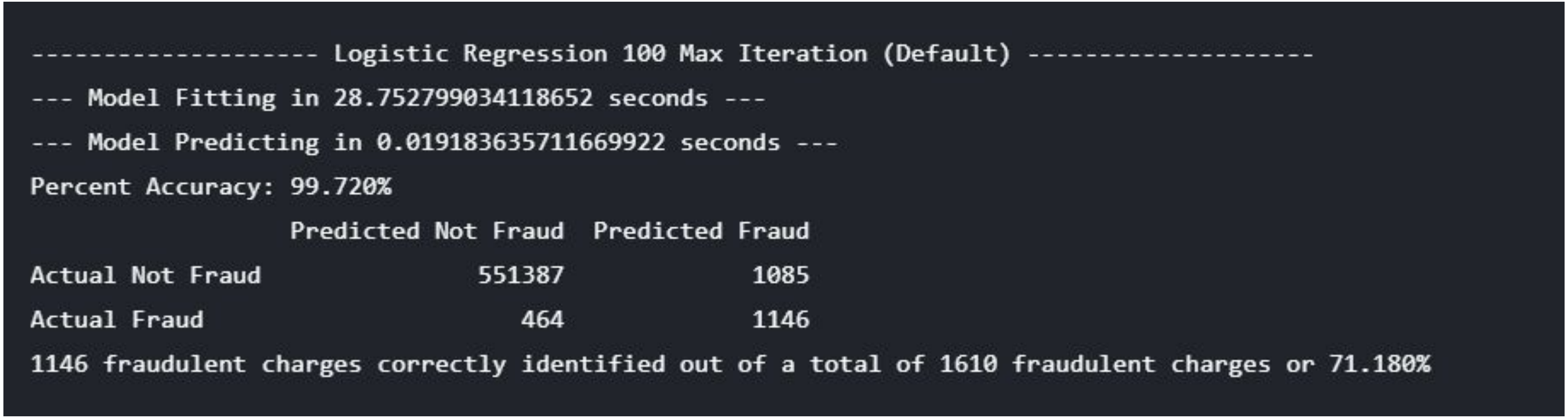
- **# of estimators**
- 50
- 100 (default)
- 200
- 500

- **Max depth**
- 1
- 3 (default)
- 5
- 7

- **Max features**
- Auto
- None(default)
- Half

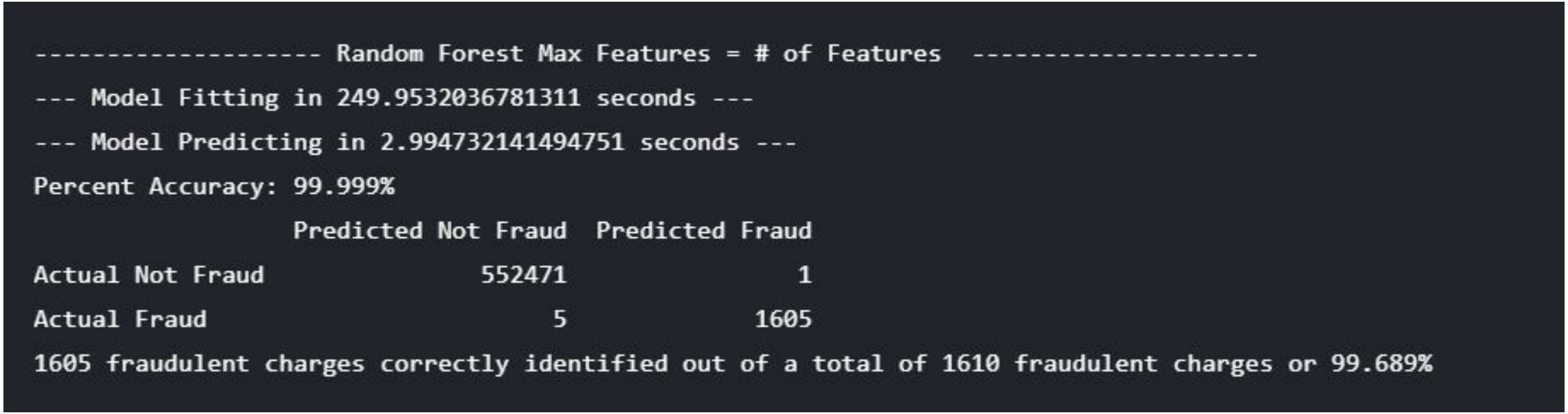
Max depth of 7 and learning rate of 0.5 drastically improve the gradient boosting model

Logistic Regression Performance



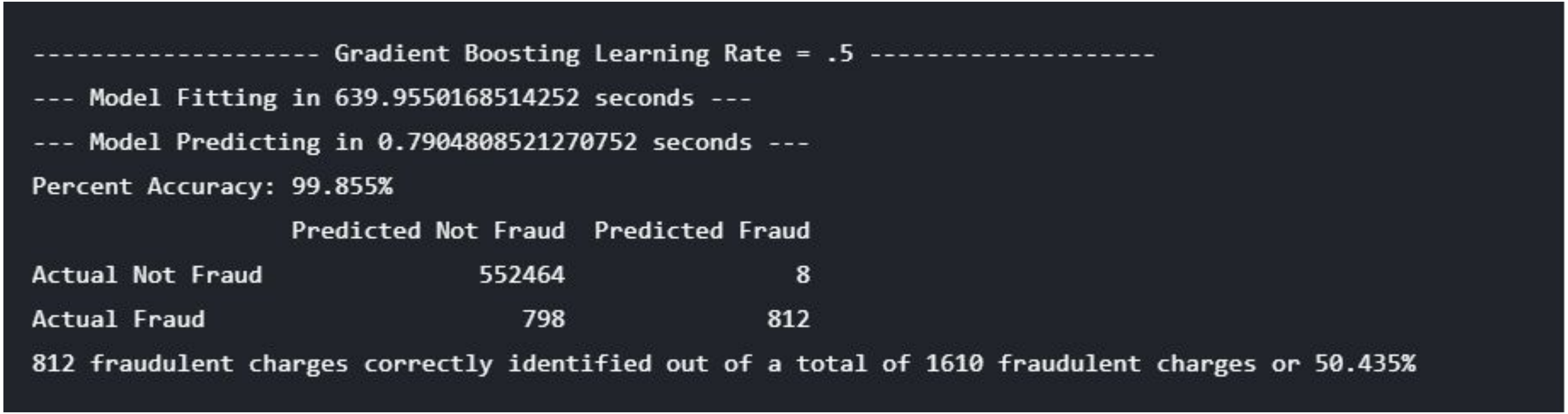
```
-------------------- Logistic Regression 100 Max Iteration (Default) --------------------
--- Model Fitting in 28.752799034118652 seconds ---
--- Model Predicting in 0.019183635711669922 seconds ---
Percent Accuracy: 99.720%
                  Predicted Not Fraud  Predicted Fraud
Actual Not Fraud               551387             1085
Actual Fraud                      464             1146
1146 fraudulent charges correctly identified out of a total of 1610 fraudulent charges or 71.180%
```

Random Forest Performance



```
-------------------- Random Forest Max Features = # of Features  --------------------
--- Model Fitting in 249.9532036781311 seconds ---
--- Model Predicting in 2.994732141494751 seconds ---
Percent Accuracy: 99.999%
                  Predicted Not Fraud  Predicted Fraud
Actual Not Fraud               552471                1
Actual Fraud                        5             1605
1605 fraudulent charges correctly identified out of a total of 1610 fraudulent charges or 99.689%
```

Gradient Boosting Performance



```
-------------------- Gradient Boosting Learning Rate = .5 --------------------
--- Model Fitting in 639.9550168514252 seconds ---
--- Model Predicting in 0.7904808521270752 seconds ---
Percent Accuracy: 99.855%
                  Predicted Not Fraud  Predicted Fraud
Actual Not Fraud               552464                8
Actual Fraud                      798              812
812 fraudulent charges correctly identified out of a total of 1610 fraudulent charges or 50.435%
```

Checking Test Sizes...

- Logistic Regression
    - 0.2 Test Sample
        - 73.70%
    - 0.3 Test Sample
        - 71.83%
    - 0.5 Test Sample
        - 71.93%

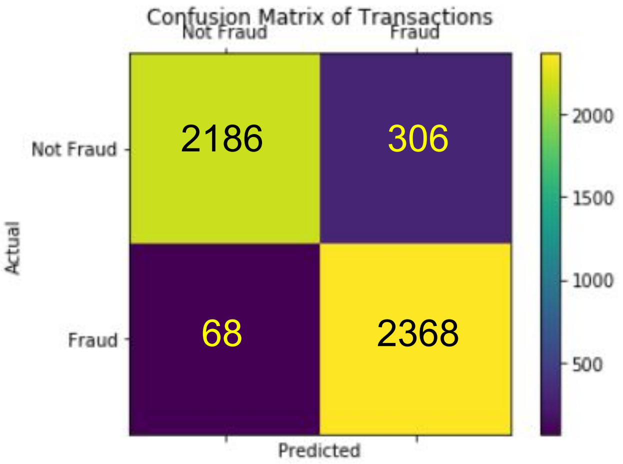
- Random Forest
    - 0.2 Test Sample
        - 99.71%
    - 0.3 Test Sample
        - 99.59%
    - 0.5 Test Sample
        - 99.46%

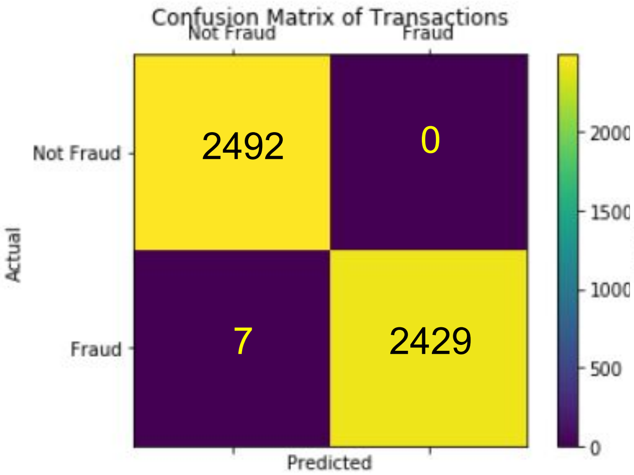
- Gradient Boosting
    - 0.2 Test Sample
        - 99.71%
    - 0.3 Test Sample
        - 99.52%
    - 0.5 Test Sample
        - 99.44%

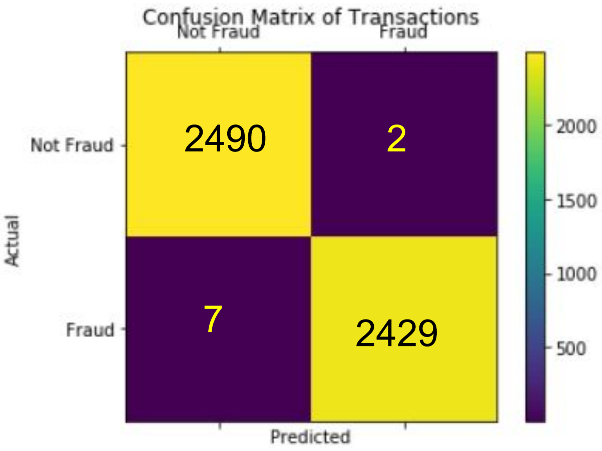
Change Sample Size

- Set number of non fraudulent transactions equal to fraudulent transactions

- Logistic Regression
  - 96.95%
- Random Forest
  - 99.55%
- Gradient Boosting
  - 99.55%

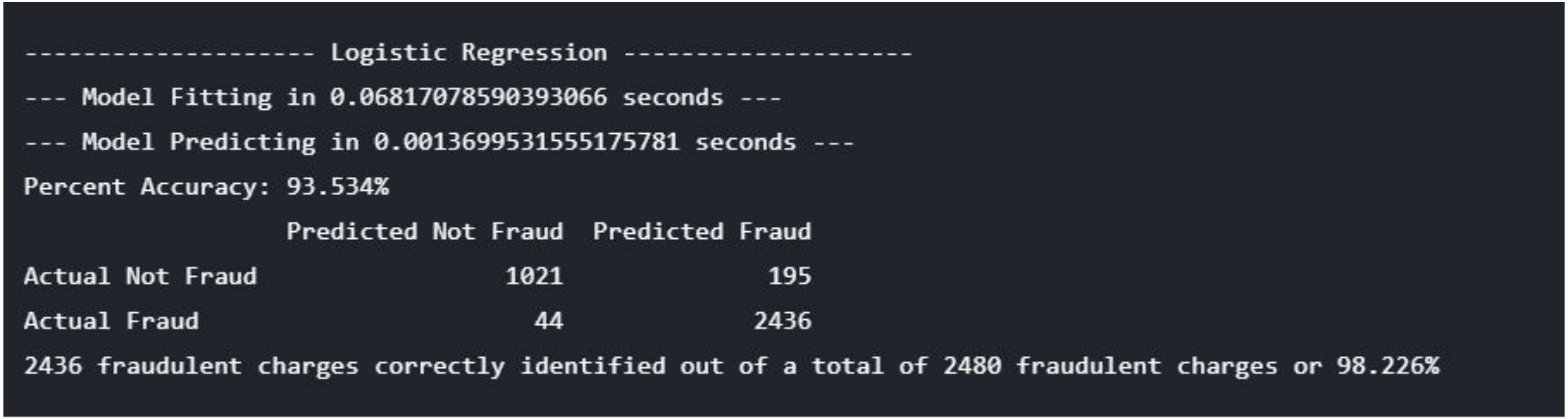






Change Sample Size Cont.

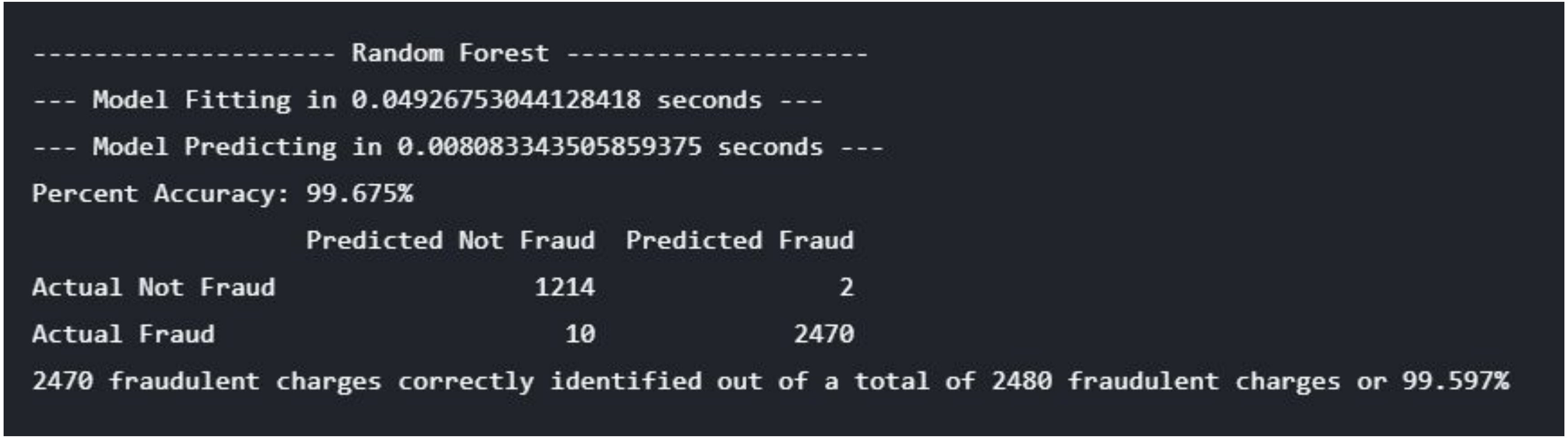
- Set number of non fraudulent transactions less than fraudulent transactions
- Logistic Regression



```
-------------------- Logistic Regression --------------------
--- Model Fitting in 0.06817078590393066 seconds ---
--- Model Predicting in 0.0013699531555175781 seconds ---
Percent Accuracy: 93.534%
                  Predicted Not Fraud  Predicted Fraud
Actual Not Fraud                 1021              195
Actual Fraud                       44             2436
2436 fraudulent charges correctly identified out of a total of 2480 fraudulent charges or 98.226%
```

Change Sample Size Cont.

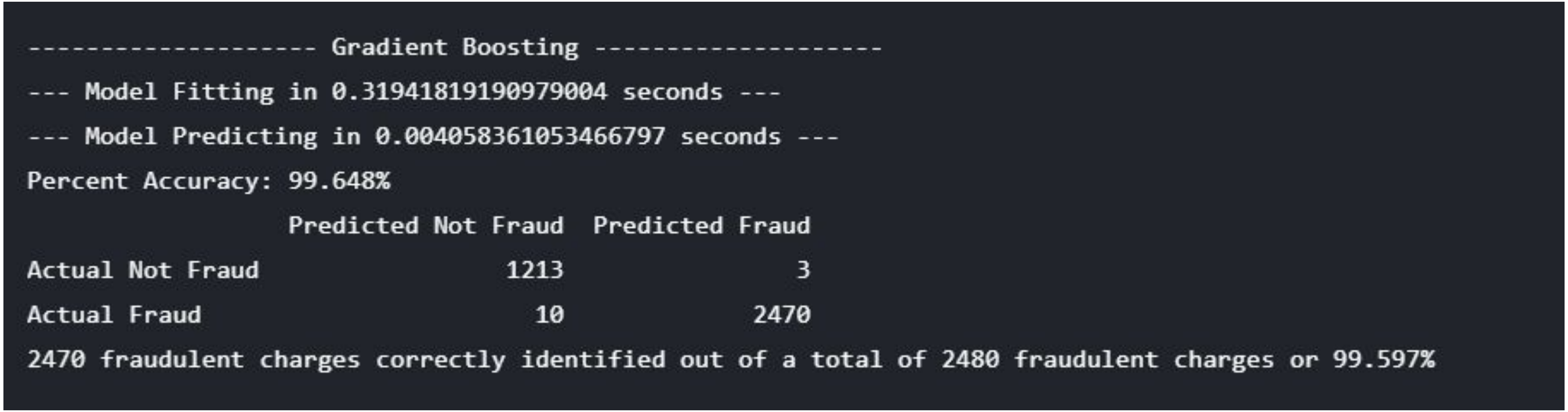
- Set number of non fraudulent transactions less than fraudulent transactions
- Random Forest



```
-------------------- Random Forest --------------------
--- Model Fitting in 0.04926753044128418 seconds ---
--- Model Predicting in 0.008083343505859375 seconds ---
Percent Accuracy: 99.675%
                  Predicted Not Fraud  Predicted Fraud
Actual Not Fraud                 1214                2
Actual Fraud                       10             2470
2470 fraudulent charges correctly identified out of a total of 2480 fraudulent charges or 99.597%
```

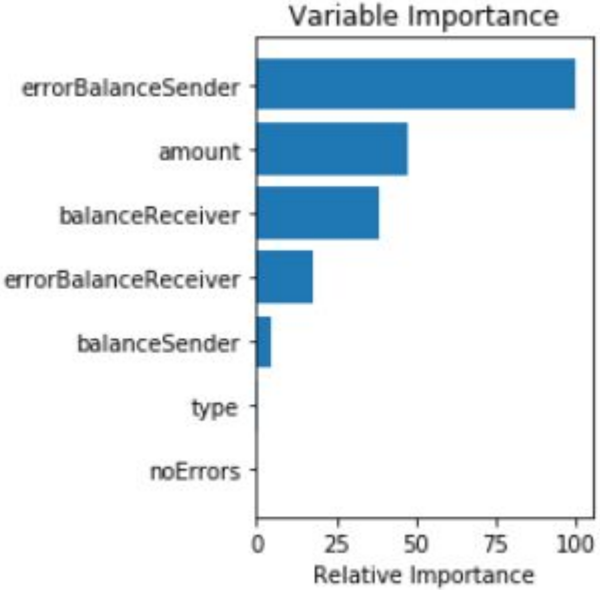
Change Sample Size Cont.

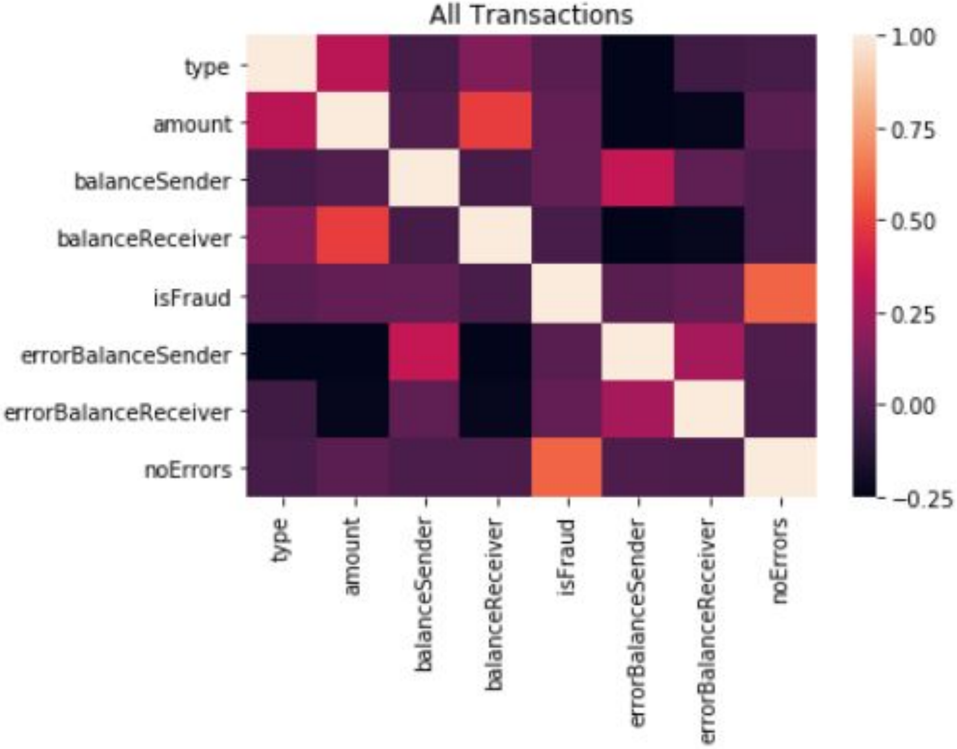
- Set number of non fraudulent transactions less than fraudulent transactions
- Gradient Boosting



```
-------------------- Gradient Boosting --------------------
--- Model Fitting in 0.31941819190979004 seconds ---
--- Model Predicting in 0.004058361053466797 seconds ---
Percent Accuracy: 99.648%
                  Predicted Not Fraud  Predicted Fraud
Actual Not Fraud                 1213                3
Actual Fraud                       10             2470
2470 fraudulent charges correctly identified out of a total of 2480 fraudulent charges or 99.597%
```

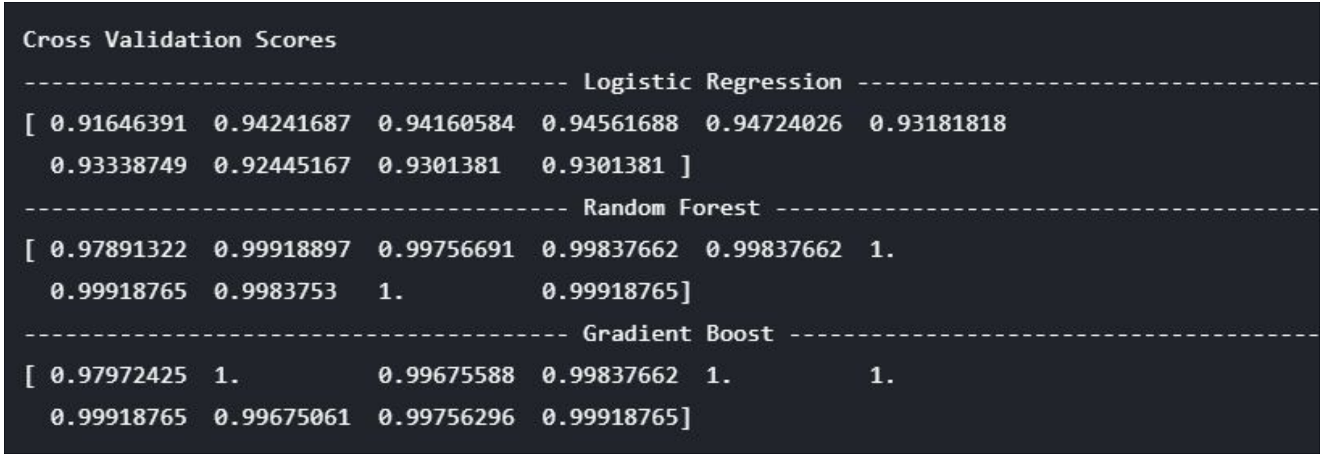
Using Gradient Boost feature importance





Using Gradient Boost feature importance

- Logistic Regression
  - 97.52%
- Random Forest
  - 99.63%
- Gradient Boosting
  - 99.63%



```
Cross Validation Scores
---------------------------------------- Logistic Regression ----------------------------------
[ 0.91646391  0.94241687  0.94160584  0.94561688  0.94724026  0.93181818
  0.93338749  0.92445167  0.9301381   0.9301381 ]
---------------------------------------- Random Forest ----------------------------------------
[ 0.97891322  0.99918897  0.99756691  0.99837662  0.99837662  1.
  0.99918765  0.9983753   1.          0.99918765]
---------------------------------------- Gradient Boost ---------------------------------------
[ 0.97972425  1.          0.99675588  0.99837662  1.          1.
  0.99918765  0.99675061  0.99756296  0.99918765]
```

Which Model to Use

Random Forest

- More accurate than Logistic Regression
    - Consistent ~99% accuracy
- Less time consuming than Gradient Boosting
    - Offers same accuracy rate as Random Forest

Questions?