Credit Card Fraud Project

SULAIMAN SALEH ALAWAD

June 26, 2019

---

1. Introduction and Overview

The dataset contains transactions made by credit cards in September 2013 by card- holders in two-day period. Of 284,807 valid transactions, 492 are listed as fraudulent. The variable 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The variable 'Amount' is the transaction value. The variable 'Class' is the response variable where 1 is a case of fraud and 0 is a valid transaction.

2. Dataset and Exploratory Analysis

The dataset for this project can be downloaded here:

https://www.kaggle.com/mlg-ulb/creditcardfraud

We will take a look on the dataset:

```
##   Time         V1          V2        V3         V4         V5         V6
## 1    0 -1.3598071 -0.07278117 2.5363467  1.3781552 -0.33832077  0.4623878
## 2    0  1.1918571  0.26615071 0.1664801  0.4481541  0.06001765 -0.0823608
## 3    1 -1.3583541 -1.34016307 1.7732093  0.3797796 -0.50319813  1.8004994
## 4    1 -0.9662717 -0.18522601 1.7929933 -0.8632913 -0.01030888  1.2472032
## 5    2 -1.1582331  0.87773675 1.5487178  0.4030339 -0.40719338  0.0959215
## 6    2 -0.4259659  0.96052304 1.1411093 -0.1682521  0.42098688 -0.0297275
##           V7         V8         V9         V10        V11        V12
## 1  0.2395985  0.0986979  0.3637870  0.09079417 -0.5515995 -0.6178009
## 2 -0.0788030  0.0851016 -0.2554251 -0.16697441  1.6127267  1.0652353
## 3  0.7914610  0.2476758 -1.5146543  0.20764287  0.6245015  0.0660837
## 4  0.2376089  0.3774359 -1.3870241 -0.05495192 -0.2264873  0.1782282
## 5  0.5929407 -0.2705327  0.8177393  0.75307443 -0.8228429  0.5381955
## 6  0.4762009  0.2603143 -0.5686714 -0.37140720  1.3412620  0.3598938
##          V13        V14       V15        V16        V17         V18
## 1 -0.9913898 -0.3111694 1.4681770 -0.4704005  0.2079712  0.02579058
## 2  0.4890950 -0.1437723 0.6355581  0.4639170 -0.1148047 -0.18336127
## 3  0.7172927 -0.1659459 2.3458649 -2.8900832  1.1099694 -0.12135931
## 4  0.5077569 -0.2879237 -0.6314181 -1.0596472 -0.6840928  1.96577500
## 5  1.3458516 -1.1196698 0.1751211 -0.4514492 -0.2370332 -0.03819479
## 6 -0.3580907 -0.1371337 0.5176168  0.4017259 -0.0581328  0.06865315
##          V19         V20          V21         V22         V23
## 1  0.4039930  0.25141210 -0.018306778  0.27783758 -0.11047391
## 2 -0.1457830 -0.06908314 -0.225775248 -0.63867195  0.10128802
## 3 -2.2618571  0.52497973  0.247998153  0.77167940  0.90941226
## 4 -1.2326220 -0.20803778 -0.108300452  0.00527360 -0.19032052
## 5  0.8034869  0.40854236 -0.009430697  0.79827849 -0.13745808
## 6 -0.0331938  0.08496767 -0.208253515 -0.55982480 -0.02639767
##          V24        V25        V26         V27         V28 Amount Class
## 1  0.0669281  0.1285394 -0.1891148  0.13355838 -0.02105305 149.62     0
## 2 -0.3398465  0.1671704  0.1258945 -0.00898310  0.01472417   2.69     0
## 3 -0.6892810 -0.3276418 -0.1390966 -0.05535279 -0.05975184 378.66     0
## 4 -1.1755753  0.6473760 -0.2219288  0.06272285  0.06145763 123.50     0
## 5  0.1412670 -0.2060096  0.5022922  0.21942223  0.21515315  69.99     0
## 6 -0.3714266 -0.2327938  0.1059148  0.25384422  0.08108026   3.67     0
```

To better understand the data we present a data dictionary of the 31 variables in the dataset.

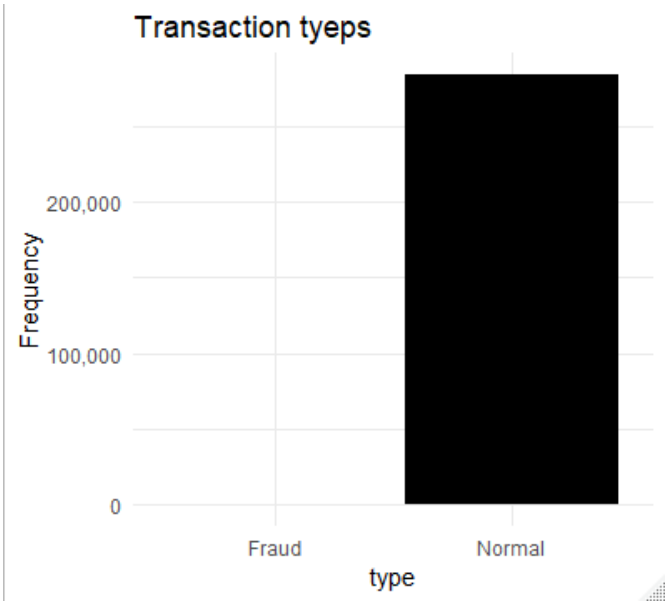
- **Time** - the number of seconds elapsed between this transaction and the first transaction in the dataset
- **V1-V28** result of a PCA Dimensionality reduction to protect user identities and sensitive features
- **Amount** - the dollar value of the transaction
- **Class** - 1 for fraudulent transactions, 0 for Normal transactions

Dimension of the dataset

Length	Columns
284807	31

We want to see how many transactions are fraudulent compared to how many are Normal. 0 is defined as a Normal transaction, and 1 is defined as a fraudulent transaction.

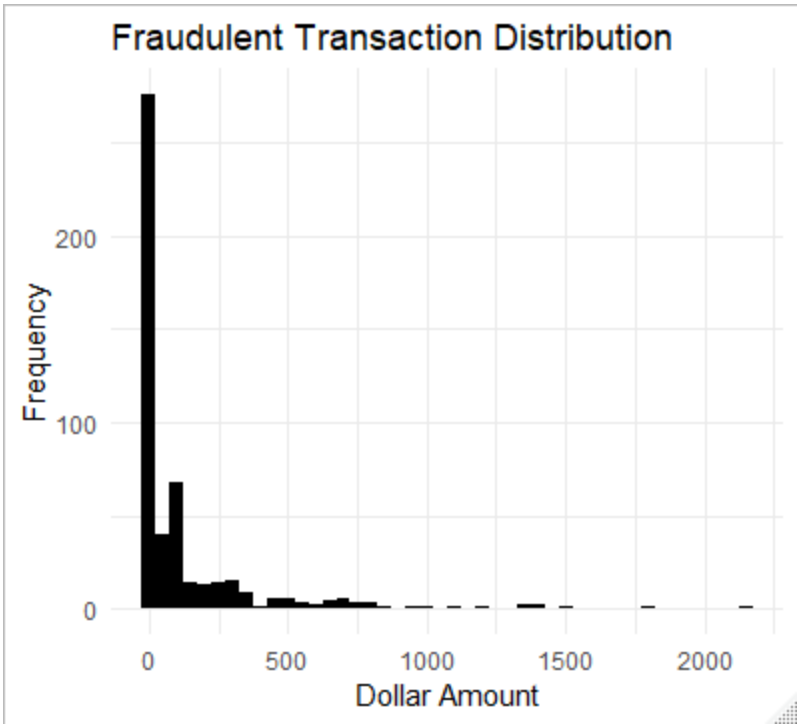
To see the data, we plot a bar graph of the frequency of fraud verses valid credit card transactions.



```
##    Min.   :-48.3256   Min.   :-5.68317   Min.   :-113.74331
##    1st Qu.: -0.8904   1st Qu.:-0.84864   1st Qu.:  -0.69160
##    Median :  0.1799   Median :-0.01985   Median :  -0.05434
##    Mean   :  0.0000   Mean   : 0.00000   Mean   :   0.00000
##    3rd Qu.:  1.0272   3rd Qu.: 0.74334   3rd Qu.:   0.61193
##    Max.   :  9.3826   Max.   :16.87534   Max.   :  34.80167
##          V6                 V7                 V8
##    Min.   :-26.1605   Min.   :-43.5572   Min.   :-73.21672
##    1st Qu.: -0.7683   1st Qu.: -0.5541   1st Qu.: -0.20863
##    Median : -0.2742   Median :  0.0401   Median :  0.02236
##    Mean   :  0.0000   Mean   :  0.0000   Mean   :  0.00000
##    3rd Qu.:  0.3986   3rd Qu.:  0.5704   3rd Qu.:  0.32735
##    Max.   : 73.3016   Max.   :120.5895   Max.   : 20.00721
##          V9                 V10                V11
##    Min.   :-13.43407   Min.   :-24.58826   Min.   :-4.79747
##    1st Qu.: -0.64310   1st Qu.: -0.53543   1st Qu.:-0.76249
##    Median : -0.05143   Median : -0.09292   Median :-0.03276
##    Mean   :  0.00000   Mean   :  0.00000   Mean   : 0.00000
##    3rd Qu.:  0.59714   3rd Qu.:  0.45392   3rd Qu.: 0.73959
##    Max.   : 15.59500   Max.   : 23.74514   Max.   :12.01891
##          V12                V13                V14
##    Min.   :-18.6837   Min.   :-5.79188   Min.   :-19.2143
##    1st Qu.: -0.4056   1st Qu.:-0.64854   1st Qu.: -0.4256
##    Median :  0.1400   Median :-0.01357   Median :  0.0506
##    Mean   :  0.0000   Mean   : 0.00000   Mean   :  0.0000
##    3rd Qu.:  0.6182   3rd Qu.: 0.66251   3rd Qu.:  0.4931
##    Max.   :  7.8484   Max.   : 7.12688   Max.   : 10.5268
##          V15                V16                V17
##    Min.   :-4.49894   Min.   :-14.12985   Min.   :-25.16280
##    1st Qu.:-0.58288   1st Qu.: -0.46804   1st Qu.: -0.48375
##    Median : 0.04807   Median :  0.06641   Median : -0.06568
##    Mean   : 0.00000   Mean   :  0.00000   Mean   :  0.00000
##    3rd Qu.: 0.64882   3rd Qu.:  0.52330   3rd Qu.:  0.39968
##    Max.   : 8.87774   Max.   : 17.31511   Max.   :  9.25353
##          V18                V19                V20
##    Min.   :-9.498746   Min.   :-7.213527   Min.   :-54.49772
##    1st Qu.:-0.498850   1st Qu.:-0.456299   1st Qu.: -0.21172
##    Median :-0.003636   Median : 0.003735   Median : -0.06248
##    Mean   : 0.000000   Mean   : 0.000000   Mean   :  0.00000
##    3rd Qu.: 0.500807   3rd Qu.: 0.458949   3rd Qu.:  0.13304
##    Max.   : 5.041069   Max.   : 5.591971   Max.   : 39.42090
##          V21                V22                V23
##    Min.   :-34.83038   Min.   :-10.933144   Min.   :-44.80774
##    1st Qu.: -0.22839   1st Qu.: -0.542350   1st Qu.: -0.16185
##    Median : -0.02945   Median :  0.006782   Median : -0.01119
##    Mean   :  0.00000   Mean   :  0.000000   Mean   :  0.00000
##    3rd Qu.:  0.18638   3rd Qu.:  0.528554   3rd Qu.:  0.14764
##    Max.   : 27.20284   Max.   : 10.503090   Max.   : 22.52841
##          V24                V25                V26
##    Min.   :-2.83663   Min.   :-10.29540   Min.   :-2.60455
##    1st Qu.:-0.35459   1st Qu.: -0.31715   1st Qu.:-0.32698
##    Median : 0.04098   Median :  0.01659   Median :-0.05214
##    Mean   : 0.00000   Mean   :  0.00000   Mean   : 0.00000
##    3rd Qu.: 0.43953   3rd Qu.:  0.35072   3rd Qu.: 0.24095
```

```
##    Max.      : 4.58455      Max.      :  7.51959      Max.      : 3.51735
##          V27                      V28                      Amount
##    Min.      :-22.565679      Min.      :-15.43008      Min.      :     0.00
##    1st Qu.: -0.070840      1st Qu.: -0.05296      1st Qu.:     5.60
##    Median :    0.001342      Median :    0.01124      Median :    22.00
##    Mean      :    0.000000      Mean      :    0.00000      Mean      :    88.35
##    3rd Qu.:    0.091045      3rd Qu.:    0.07828      3rd Qu.:    77.17
##    Max.      : 31.612198      Max.      : 33.84781      Max.      :25691.16
##          Class
##    Min.      :0.000000
##    1st Qu.:0.000000
##    Median :0.000000
##    Mean      :0.001728
##    3rd Qu.:0.000000
##    Max.      :1.000000
```

We want to investigate the dollar amounts of fraud. Here we plot all the fraudulent transaction by amount. This plot shows a massive skew toward transactions under $100.



Amount	count
1.00	113
0.00	27
99.99	27
0.76	17
0.77	10
0.01	5
2.00	4
3.79	4
0.68	3
1.10	3

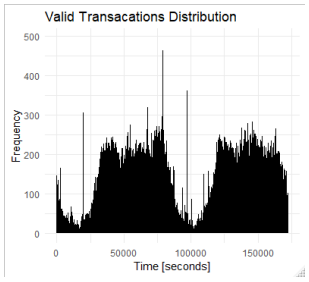
We can also investigate what are the most common valid transactions in the dataset.

Amount	count
1.00	13575
1.98	6044
0.89	4872
9.99	4746
15.00	3280
0.76	2981
10.00	2950
1.29	2892
1.79	2622
0.99	2304

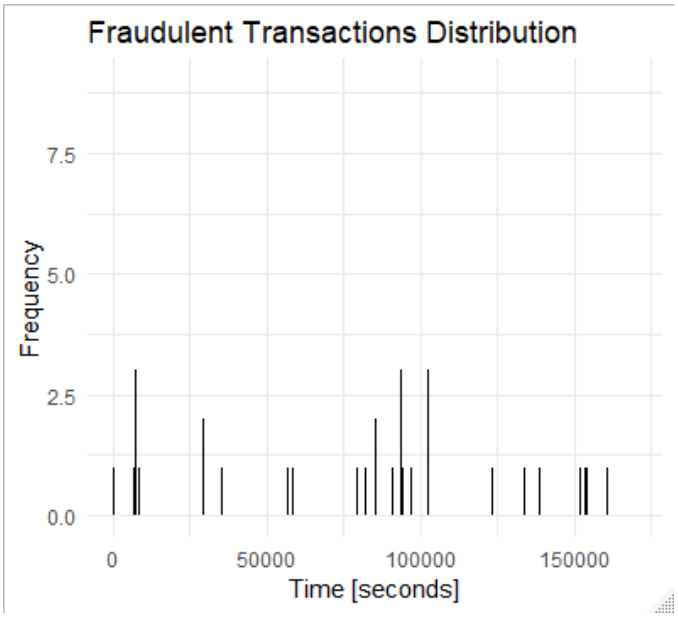
An interesting observation is that $1 is the most common fraudulent and valid transaction. In fact the chance of a transaction of $1 being fraud is almost five times higher than other transactions in the data set.

Another very interesting observations is that a transactions of $99.99 is the 98th most common valid trans- actions with 303 transactions, but is tied for second of fraudulent transactions with 27. This means that ~9% of $99.99 transactions in the data set are fraudulent!

We can plot a distribution of valid transactions over time. This plot has a clear episodic distribution. This makes sense since a day has 86,400 seconds, which is the approximate period of this distribution. The punchline is that most transactions occur during the day, while fewer transactions occur at night. There is a clear spike of outlier transactions near the trough of the graph. We surmise that these spikes correlate to automated transactions that are processed a little before the close of midnight or shortly after midnight. An example of automated transactions would be monthly recurring bills set to autopay.



Similarly, to the distribution of valid transactions, we can plot the distribution of fraudulent transactions over time. The lack of any clear episodic distribution indicates that fraud can occur at any time.



To note: Without performing Fourier analysis (such as a Fast Fourier Transform) on this data, we do not know with certainty that fraudulent transactions are non-episodic. This analysis is beyond the scope of this project, and the frequency distribution plotted above will suffice to show that fraudulent transactions are not episodic and can occur at any point in time.

We want to calculate the correlation between the variables and graph them. We first design a correlation matrix.

Here is a matrix of the correlation between the 31 distinct variables.

	V28	V27	V26	V25	V24	V20	V17	V15	V14	V13	V9	V8	V6	V4	V2	V3	V23	V22	V21	V19	V18	V16	V12	V10	V1	Time	V5	V7	Amount	V11	Class
V28	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	0.00	0.01	0.00	0.01
V27	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	0.00	0.03	0.00	0.02
V26	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.04	0.00	0.00	0.00	0.00	0.00
V25	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.23	0.00	0.00	-0.05	0.00	0.00
V24	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.02	0.00	0.00	0.01	0.00	-0.01
V20	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.05	0.00	0.00	0.34	0.00	0.02
V17	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.07	0.00	0.00	0.01	0.00	-0.33
V15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.18	0.00	0.00	0.00	0.00	0.00
V14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.10	0.00	0.00	0.03	0.00	-0.30
V13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.07	0.00	0.00	0.01	0.00	0.00
V9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	0.00	-0.04	0.00	-0.10
V8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.04	0.00	0.00	-0.10	0.00	0.02
V6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.06	0.00	0.00	0.22	0.00	-0.04
V4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.11	0.00	0.00	0.10	0.00	0.13
V2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00	0.00	-0.53	0.00	0.09
V3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.42	0.00	0.00	-0.21	0.00	-0.19
V23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	-0.11	0.00	0.00
V22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.00	-0.06	0.00	0.00
V21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.11	0.00	0.04
V19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	-0.06	0.00	0.03
V18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.04	0.00	-0.11
V16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	-0.20
V12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.12	0.00	0.00	-0.01	0.00	-0.26
V10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.03	0.00	0.00	-0.10	0.00	-0.22
V1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.12	0.00	0.00	-0.23	0.00	-0.10
Time	-0.01	-0.01	-0.04	-0.23	-0.02	-0.05	-0.07	-0.18	-0.10	-0.07	-0.01	-0.04	-0.06	-0.11	-0.01	-0.42	0.05	0.14	0.04	0.03	0.09	0.01	0.12	0.03	0.12	1.00	0.17	0.08	-0.01	-0.25	-0.01
V5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.17	1.00	0.00	-0.39	0.00	-0.09
V7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	1.00	0.40	0.00	-0.19
Amount	0.01	0.03	0.00	-0.05	0.01	0.34	0.01	0.00	0.03	0.01	-0.04	-0.10	0.22	0.10	-0.53	-0.21	-0.11	-0.06	0.11	-0.06	0.04	0.00	-0.01	-0.10	-0.23	-0.01	-0.39	0.40	1.00	0.00	0.01
V11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	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	0.15
Class	0.01	0.02	0.00	0.00	-0.01	0.02	-0.33	0.00	-0.30	0.00	-0.10	0.02	-0.04	0.13	0.09	-0.19	0.00	0.00	0.04	0.03	-0.11	-0.20	-0.26	-0.22	-0.10	-0.01	-0.09	-0.19	0.01	0.15	1.00

Further, we can plot the correlation. Notice how all the variables V1-V28 have very low correlation coefficients among each other, and especially low correlation with the ‘Class’ feature. This was already expected since the data was processed using PCA.

we established that fraud does not appear to coincide with a specific time of day, so the ‘Time’ variable will be removed from the dataset.

To verify the variable ‘Time’ has been removed, we can view the first six entries with the head() function.

##	V1	V2	V3	V4	V5	V6
## 1	**-1.3598071**	**-0.07278117**	**2.5363467**	**1.3781552**	**-0.33832077**	**0.46238778**
## 2	**1.1918571**	**0.26615071**	**0.1664801**	**0.4481541**	**0.06001765**	**-0.08236081**
## 3	**-1.3583541**	**-1.34016307**	**1.7732093**	**0.3797796**	**-0.50319813**	**1.80049938**
## 4	**-0.9662717**	**-0.18522601**	**1.7929933**	**-0.8632913**	**-0.01030888**	**1.24720317**
## 5	**-1.1582331**	**0.87773675**	**1.5487178**	**0.4030339**	**-0.40719338**	**0.09592146**
## 6	**-0.4259659**	**0.96052304**	**1.1411093**	**-0.1682521**	**0.42098688**	**-0.02972755**

##	V7	V8	V9	V10	V11	V12
## 1	0.23959855	0.09869790	0.3637870	0.09079417	-0.5515995	-0.61780086
## 2	-0.07880298	0.08510165	-0.2554251	-0.16697441	1.6127267	1.06523531
## 3	0.79146096	0.24767579	-1.5146543	0.20764287	0.6245015	0.06608369
## 4	0.23760894	0.37743587	-1.3870241	-0.05495192	-0.2264873	0.17822823
## 5	0.59294075	-0.27053268	0.8177393	0.75307443	-0.8228429	0.53819555
## 6	0.47620095	0.26031433	-0.5686714	-0.37140720	1.3412620	0.35989384
##	V13	V14	V15	V16	V17	V18
## 1	0.9913898	0.3111694	1.4681770	-0.4704005	0.20797124	0.02579058
## 2	0.4890950	-0.1437723	0.6355581	0.4639170	-0.11480466	-0.18336127
## 3	0.7172927	-0.1659459	2.3458649	-2.8900832	1.10996938	-0.12135931
## 4	0.5077569	-0.2879237	-0.6314181	-1.0596472	-0.68409279	1.96577500
## 5	1.3458516	-1.1196698	0.1751211	-0.4514492	-0.23703324	-0.03819479
## 6	-0.3580907	-0.1371337	0.5176168	0.4017259	-0.05813282	0.06865315
##	V19	V20	V21	V22	V23	V24
## 1	0.40399296	0.25141210	-0.018306778	0.277837576	-0.11047391	0.06692807
## 2	-0.14578304	-0.06908314	-0.225775248	-0.638671953	0.10128802	-0.33984648
## 3	-2.26185710	0.52497973	0.247998153	0.771679402	0.90941226	-0.68928096
## 4	-1.23262197	-0.20803778	-0.108300452	0.005273597	-0.19032052	-1.17557533
## 5	0.80348692	0.40854236	-0.009430697	0.798278495	-0.13745808	0.14126698
## 6	-0.03319379	0.08496767	-0.208253515	-0.559824796	-0.02639767	-0.37142658

##	V25	V26	V27	V28	Amount	Class
## 1	0.1285394	-0.1891148	0.133558377	-0.02105305	149.62	0
## 2	0.1671704	0.1258945	-0.008983099	0.01472417	2.69	0
## 3	-0.3276418	-0.1390966	-0.055352794	-0.05975184	378.66	0
## 4	0.6473760	-0.2219288	0.062722849	0.06145763	123.50	0
## 5	-0.2060096	0.5022922	0.219422230	0.21515315	69.99	0
## 6	-0.2327938	0.1059148	0.253844225	0.08108026	3.67	0

II. Methods and Analysis

For this report we will investigate four models: The Naive Model, the Naive Bayes Model, the K-Nearest Neighbor Model, and the Random Forest Model.

III.A. Naive Model

The first model we design is the Naive Model. This model makes the simple prediction that every transaction is a valid transaction and that there are no fraudulent transactions. This will serve as our first attempt in trying to better the model.

III.B. Naive Bayes Model

The Naive Bayes Model is a model that applies Bayes' theorem with strong (naive) independence assumptions between the features. We build the model with the 'Class' (i.e. whether the transaction is valid or fraud) as the target and with the remaining variables are predictors.

III.C. K-Nearest Neighbor

The K-Nearest Neighbors algorithm (KNN) is a non-parametric method used for classification where the input consists of the k closest training examples in the feature space. In KNN classification (determining if the transaction was valid or fraud), the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. Several values of k were tested and 5 was chosen as a value that provided the best results. In this model, 'Class' is the target and all other variables are predictors.

III.D. Random Forest

The Random Forest algorithm (sometimes called Random Decision Forests) is an algorithm of machine learning were an ensemble learning method for classification operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classification of the individual trees. These trees are a decision tree that goes from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). For

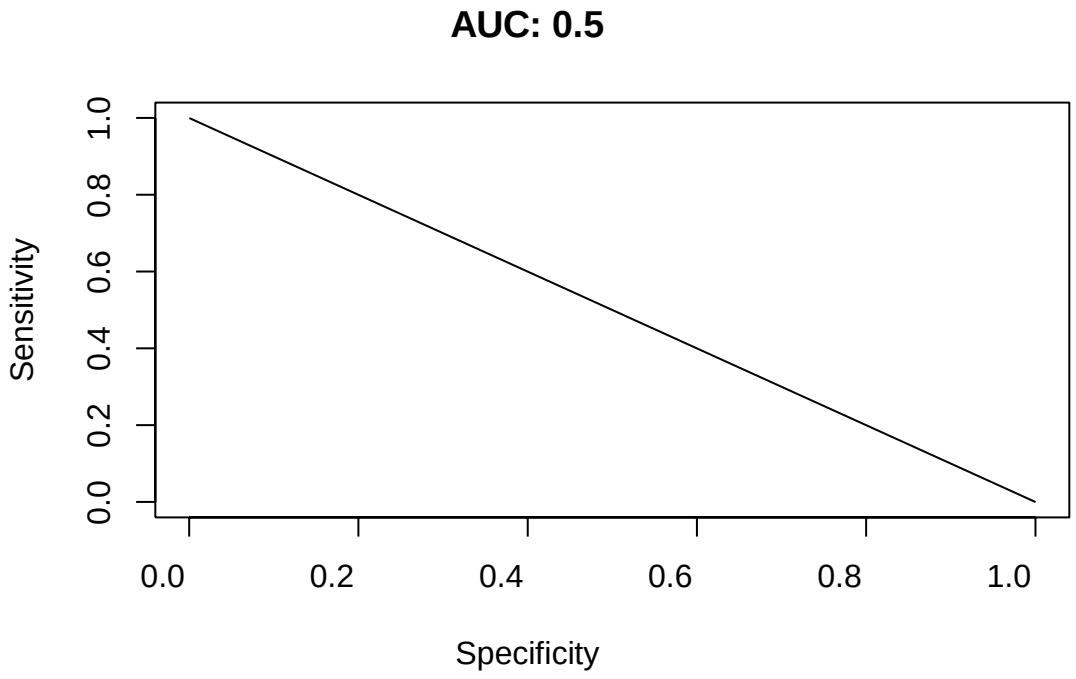
this model, ‘Class’ (whether a transaction is valid or fraud) is the target, and all other variables are predictors. In this model we define the number of trees to be 500.

III. Results

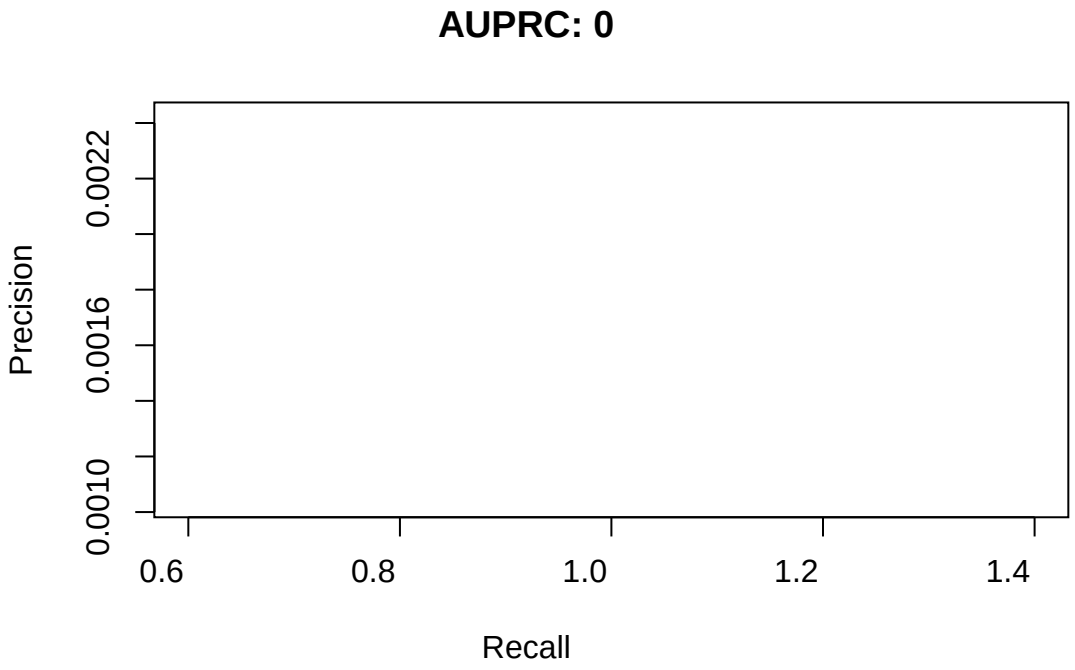
Prior to our computations, we partition the dataset into a training set, a test set, and a cross validation set.

IV.A. Naive Model

When we plot sensitivty and specificty for the Naive Model, we obtain a straight diagonal line, as expected. The Area Under the Curve (AUC) for this model yeilds 0.5.



No line is generated for the Area Under the Precision Recall Curve (AUPRC) since these values are zero.



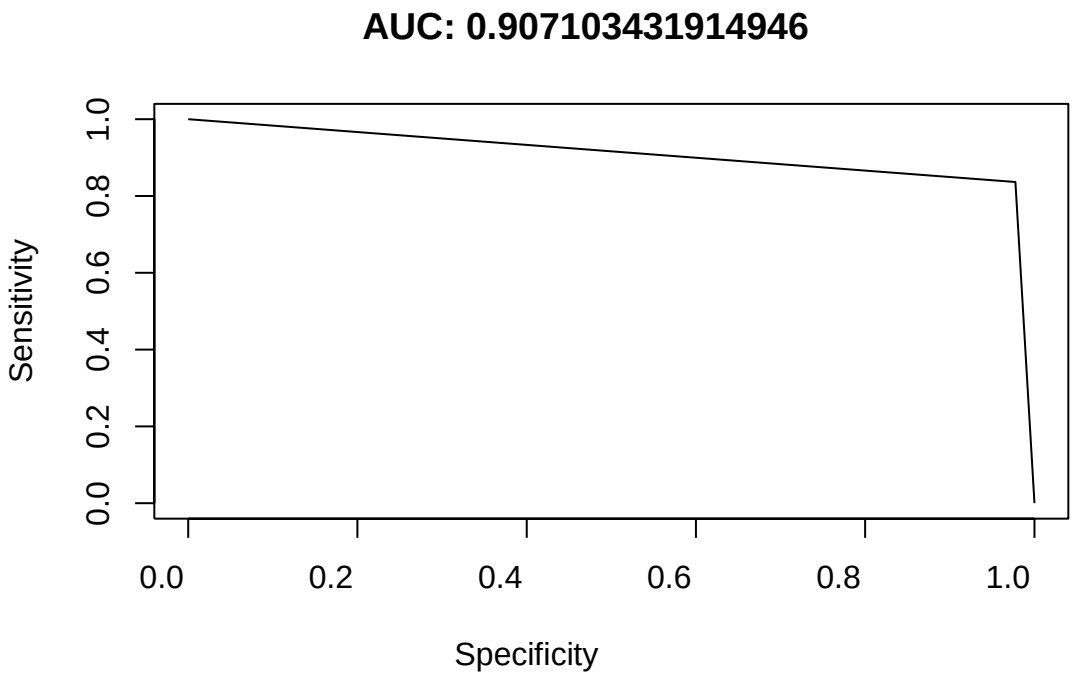
We save our results from our first model in a data frame and display them.

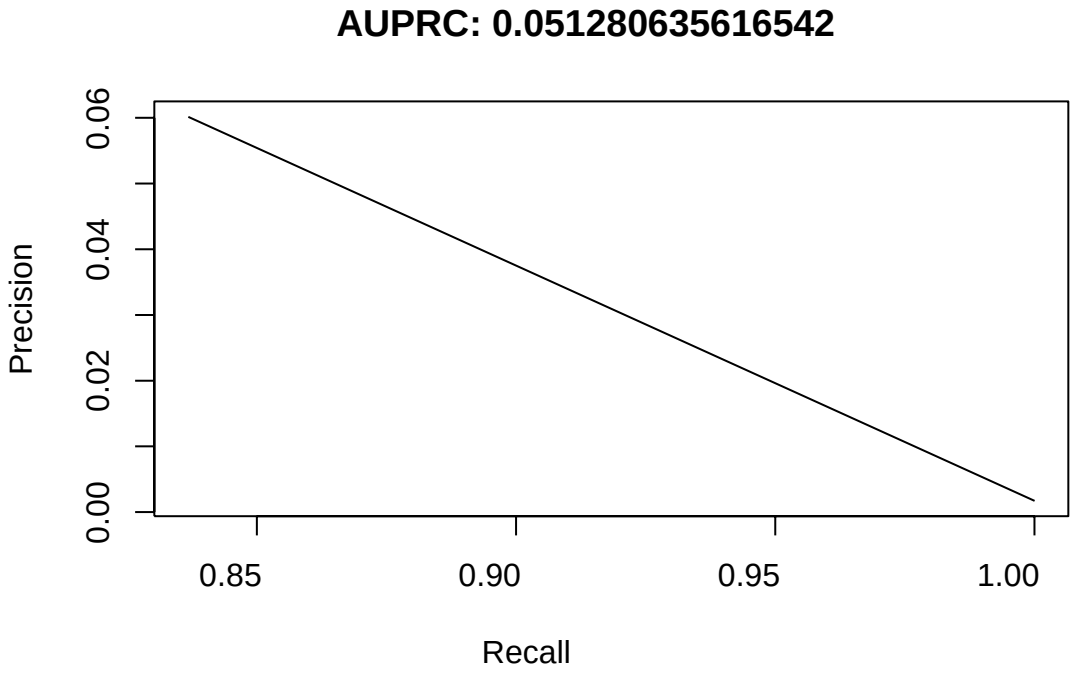
Model	AUC	AUPRC
Naive	0.5	0

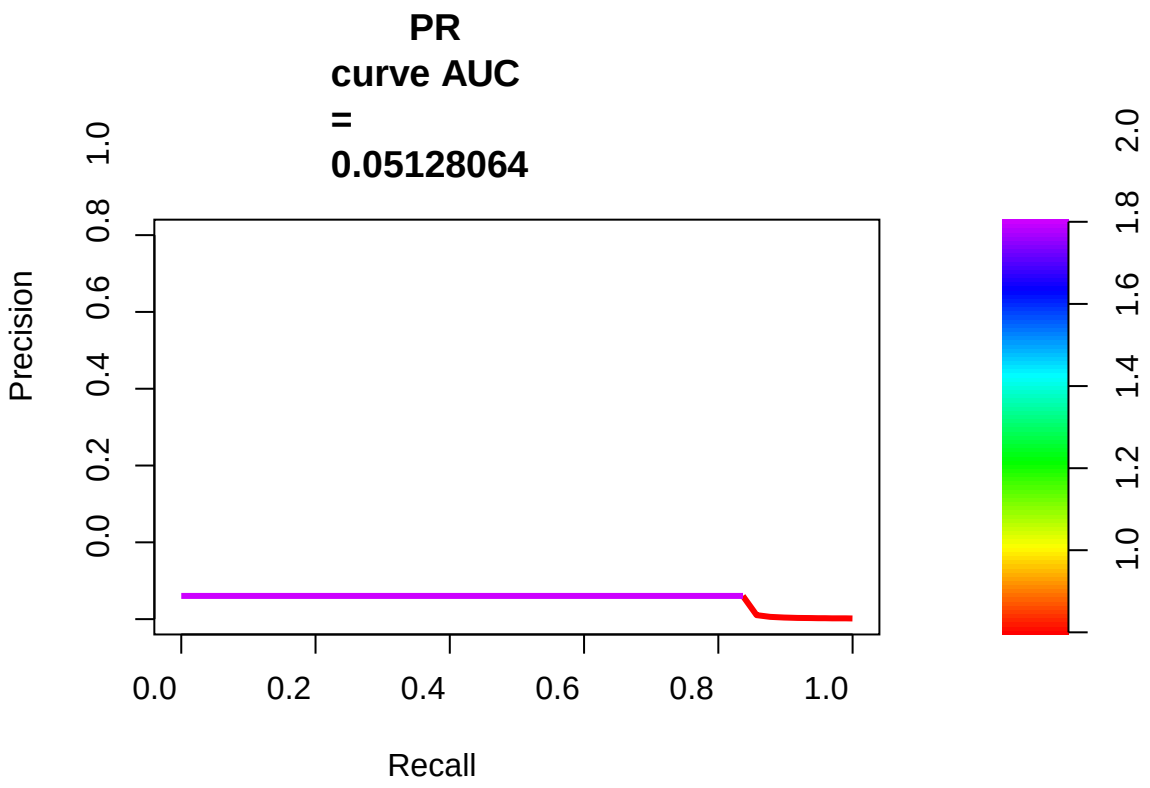
Although this model has an accuracy ~99.8%, it has a AUPRC of 0, and therefore is comepletely useless for our task at hand.

IV.B. Naive Bayes Model

The AUC for sensitivity versus specificity for the Naive Bayes Model is greatly improved compared to the Naive Model alone. Additionally, the AUPRC improves (albeit marginally) to just 0.05. We can improve on this with the following two models.


AUC: 0.907103431914946
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
Specificity


AUPRC: 0.051280635616542
Precision
0.00
0.02
0.04
0.06
0.85
0.90
0.95
1.00
Recall

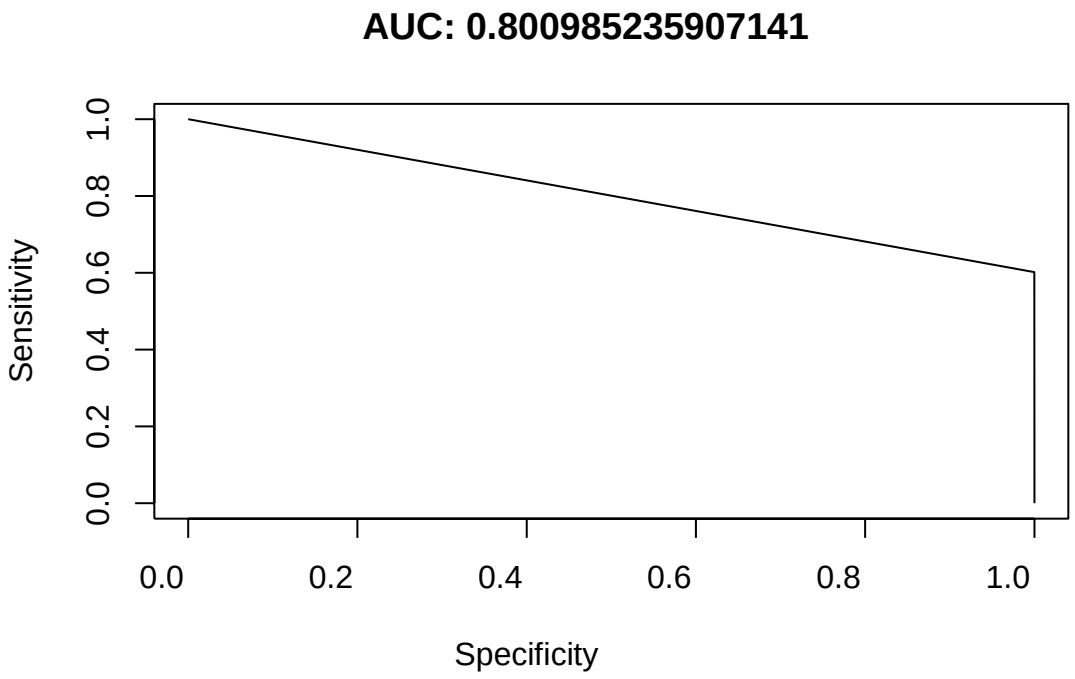


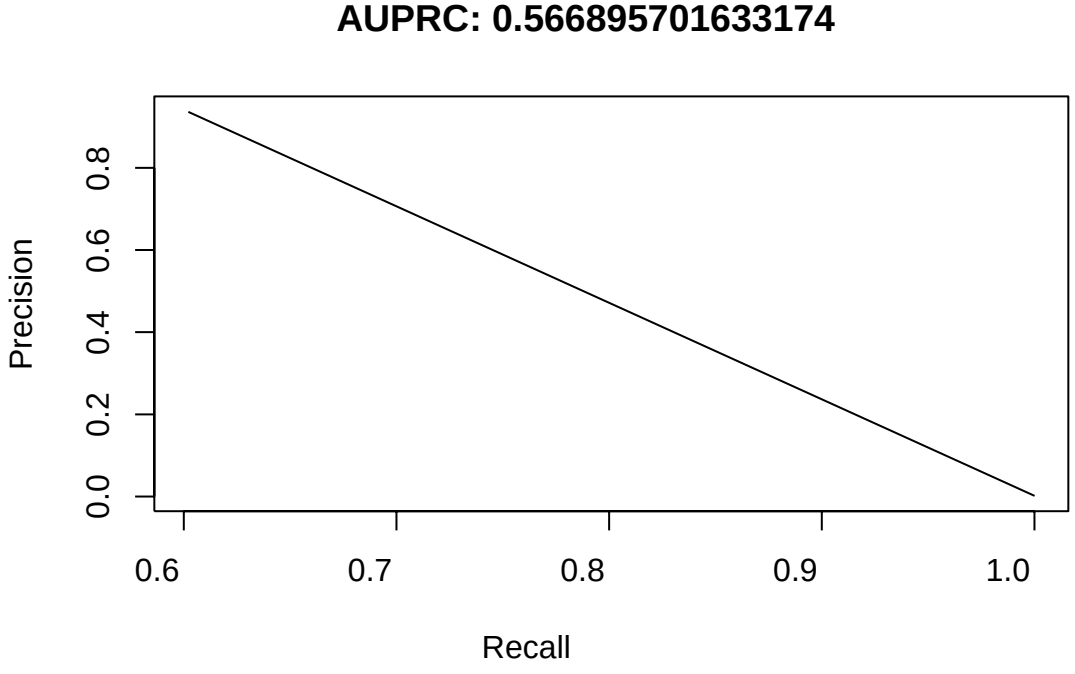
We save our results from our Naive Bayes Model in a data frame and display them with previous results.

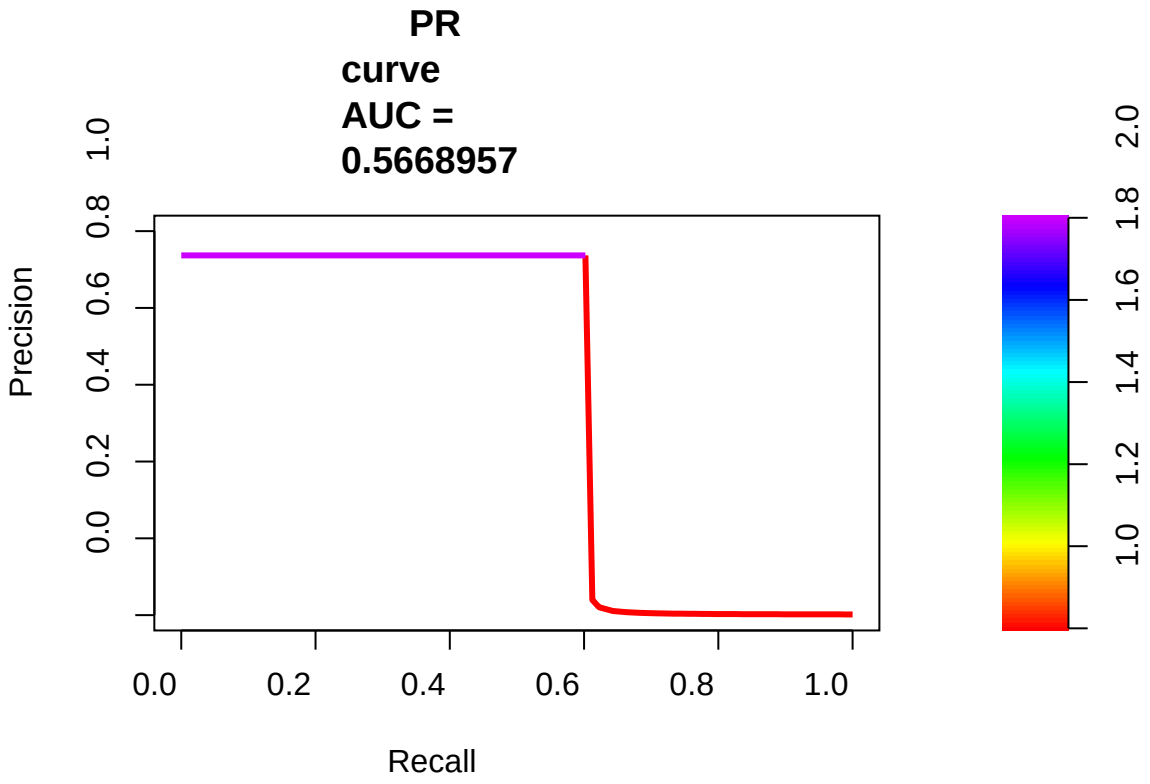
Model	AUC	AUPRC
Naive	0.5000000	0.0000000
Naive Bayes	0.9071034	0.0512806

IV.C. K-Nearest Neighbor

For the K Nearest Neighbors, we have a small reduction for our AUC when looking at sensitivity versus specificity compared to the Naive Bayes Model, but a substantial improvement on precision versus recall in our AUPRC. This value of 0.56 is still low. We would like to achieve an AUC for precision versus recall close to 0.8.


AUC: 0.800985235907141
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
Specificity


AUPRC: 0.566895701633174
Precision
0.0
0.2
0.4
0.6
0.8
0.6
0.7
0.8
0.9
1.0
Recall

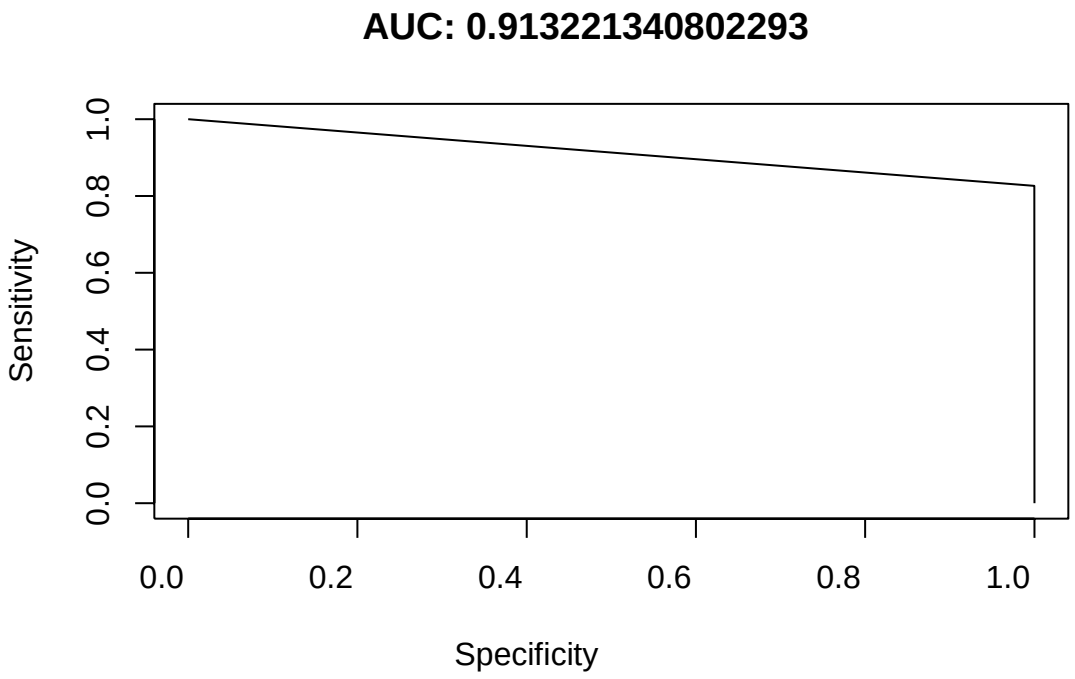


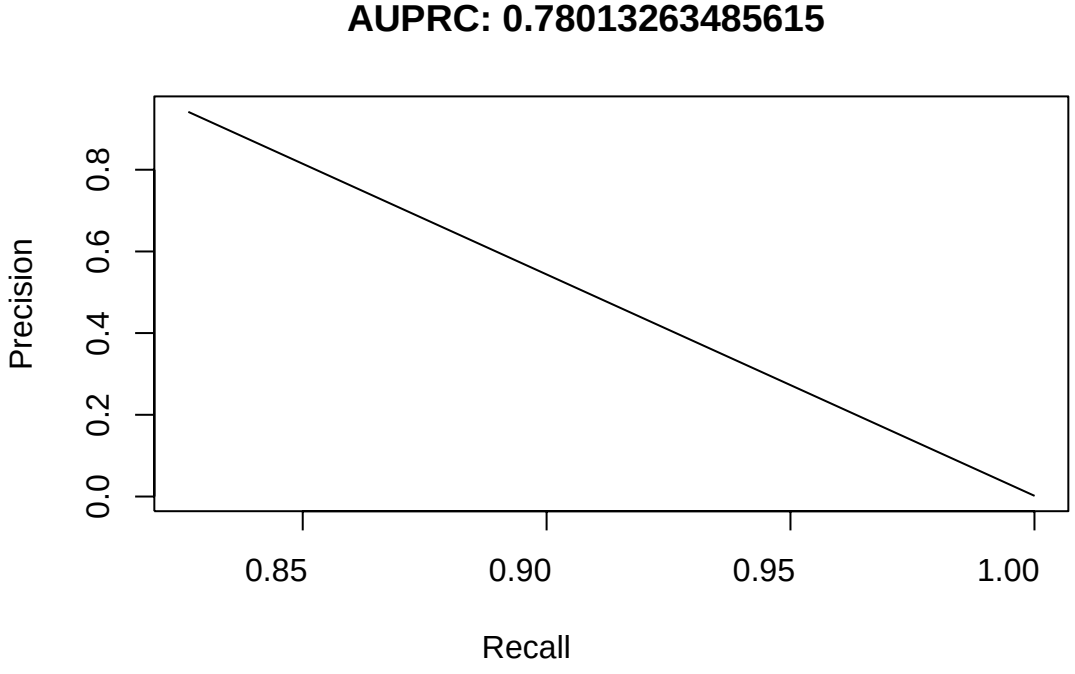
We save our results from our K-Nearest Neighbor Model in a data frame and display them with previous results.

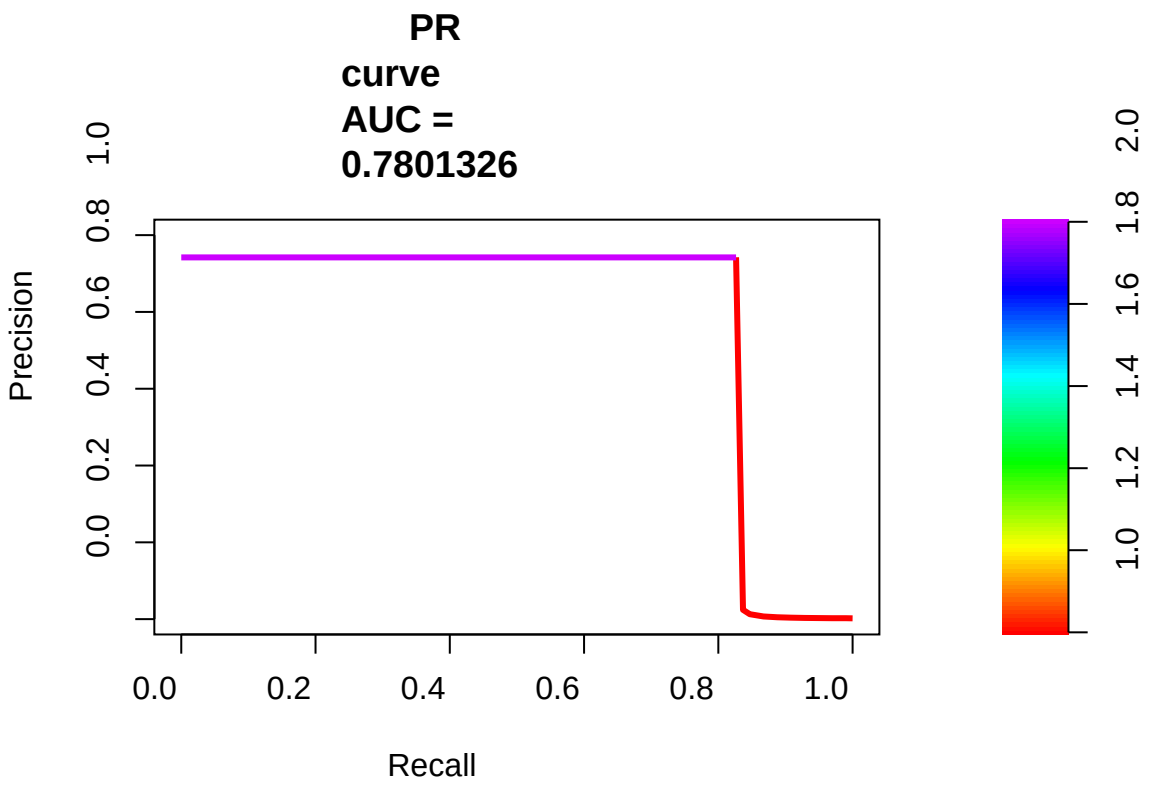
Model	AUC	AUPRC
Naive	0.5000000	0.0000000
Naive Bayes	0.9071034	0.0512806
K-Nearest Neighbors	0.8009852	0.5668957

IV.D. Random Forest

For our Random Forest Model, we not only obtain the best AUC for sensitivity versus specificity (0.91), but we also obtain the best AUC for precision versus recall (0.78). Out of the models developed and trained, this model is the most accurate for our task at hand. The use of 500 trees for this algorithm worked well.


AUC: 0.913221340802293
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
Specificity


AUPRC: 0.78013263485615
Precision
0.0
0.2
0.4
0.6
0.8
0.85
0.90
0.95
1.00
Recall



We save our results from our Random Forest Model in a data frame and display them with previous results.

Model	AUC	AUPRC
Naive	0.5000000	0.0000000
Naive Bayes	0.9071034	0.0512806
K-Nearest Neighbors	0.8009852	0.5668957
Random Forest	0.9132213	0.7801326

IV. Conclusion

In this report we seek to address credit card fraud using a machine learning approach. Since credit card fraud is very rare compared to the volume of valid transactions, we are posed with a machine learning problem that utilizes the accuracy of the model by calculating the Area Under the Precision-Recall Curve as opposed to a more traditional method such as a confusion matrix.

Four models were developed and each was tested with a dataset of credit card transactions provided by Kaggle. Here we again present the findings from the four models utilized for this report.

Model	AUC	AUPRC
Naive	0.5000000	0.0000000
Naive Bayes	0.9071034	0.0512806
K-Nearest Neighbors	0.8009852	0.5668957
Random Forest	0.9132213	0.7801326

The end