Data Science Career Track

Capstone 1 -

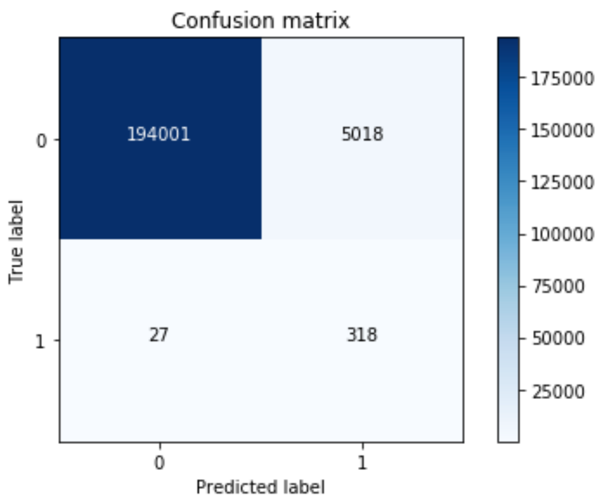
Milestone Report

by Edward Franke

05/01/2019

Credit Card Fraud Detection Project – Milestone Report

EXECUTIVE SUMMARY:
The purpose for this project is to find a correlation connected to fraudulent credit card transactions that can separate them in real time compared to legitimate transactions. The dataset is a cleaned dataset from Kaggle.com. It has been discovered there is a correlation connected to fraudulent transactions. Most models fail to accurate predict which transactions are fraudulent because the dataset is severely unbalanced with over 99% of the transactions being legitimate. The solution was to run SMOTE first to rebalance the dataset then logistic regression with these results:



CONCEPTS:
IDEA: A model to detect fraud in credit card transactions. (problem to solve)
CLIENT: Credit Card Companies
REASON: If they don't detect and stop the fraud as it happens, their customer don't pay the cost, they do. This project is intended to reduce their expenses and increase their profits.
DATA: From Kaggle, a cleaned dataset with 284,807 transactions from 2013 with 492 frauds in total.
SOLUTION: Create a model or analysis to discover what makes fraudulent transaction similar to detect this relationship as it happens.
DETAILS: See Detail Section Below.
DELIVERABLES: Working code that detect fraud as it receives data and a presentation outlining the discoveries and explain the methods used to reach the discoveries.

INITIAL FINDINGS FROM EXPLORATORY ANALYSIS:

```
data_fraud_df = data_df[data_df.Class == 1]
data_legit_df = data_df[data_df.Class == 0]
```

This is the code that was used to separate the initial dataframe into separate dataframes only containing fraud or legit transactions.

While comparing the transactions, I noticed fraudulent transactions have a greater histogram area that legitimate transactions. I was under the impression that I could create code to determine any transaction with a total V set as below -17 as fraudulent (see Appendix for more details). However, as this overlayed histogram comparison shows, that would fail the purpose of the project. A solution is still being sought.

Fraudulent Transactions (Class 1 indicated fraud)

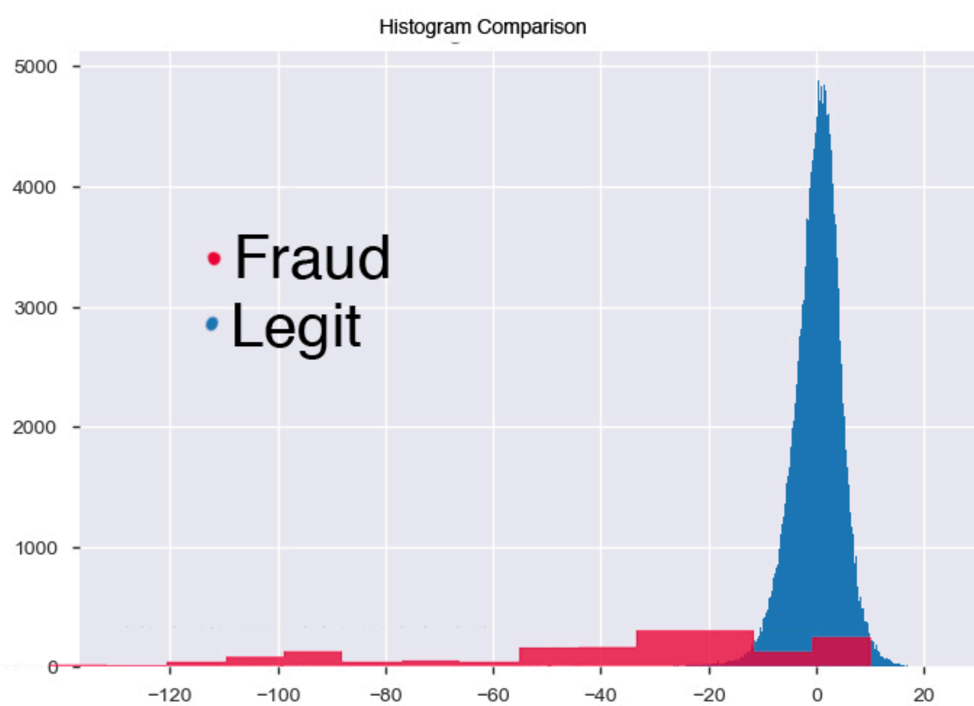
Warnings

- `Amount` has 27 / 5.5% zeros Zeros
- `Class` has constant value 1 Rejected
- `Row_Mean` is highly correlated with `Total` (ρ = 1) Rejected
- `Time` is highly correlated with `index` (ρ = 0.99465) Rejected
- `Total` is highly correlated with `V10` (ρ = 0.94572) Rejected
- `V17` is highly correlated with `V16` (ρ = 0.96015) Rejected
- `V18` is highly correlated with `V17` (ρ = 0.97149) Rejected
- `V3` is highly correlated with `V1` (ρ = 0.90788) Rejected

Legitimate Transactions (Class 0 indicates legit)

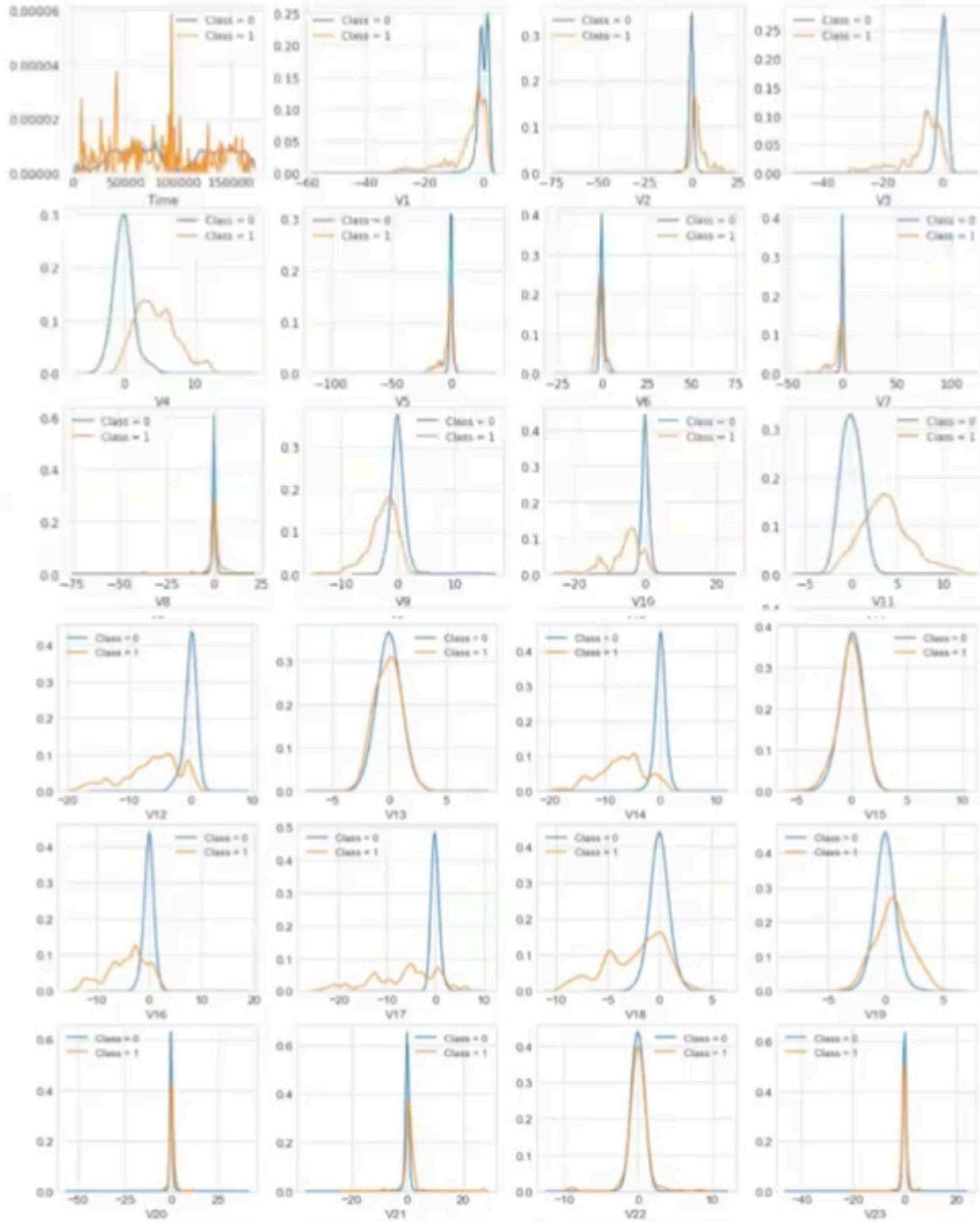
Warnings

- `Class` has constant value 0 Rejected
- `Row_Mean` is highly correlated with `Total` (ρ = 1) Rejected
- `Time` is highly correlated with `index` (ρ = 0.99338) Rejected

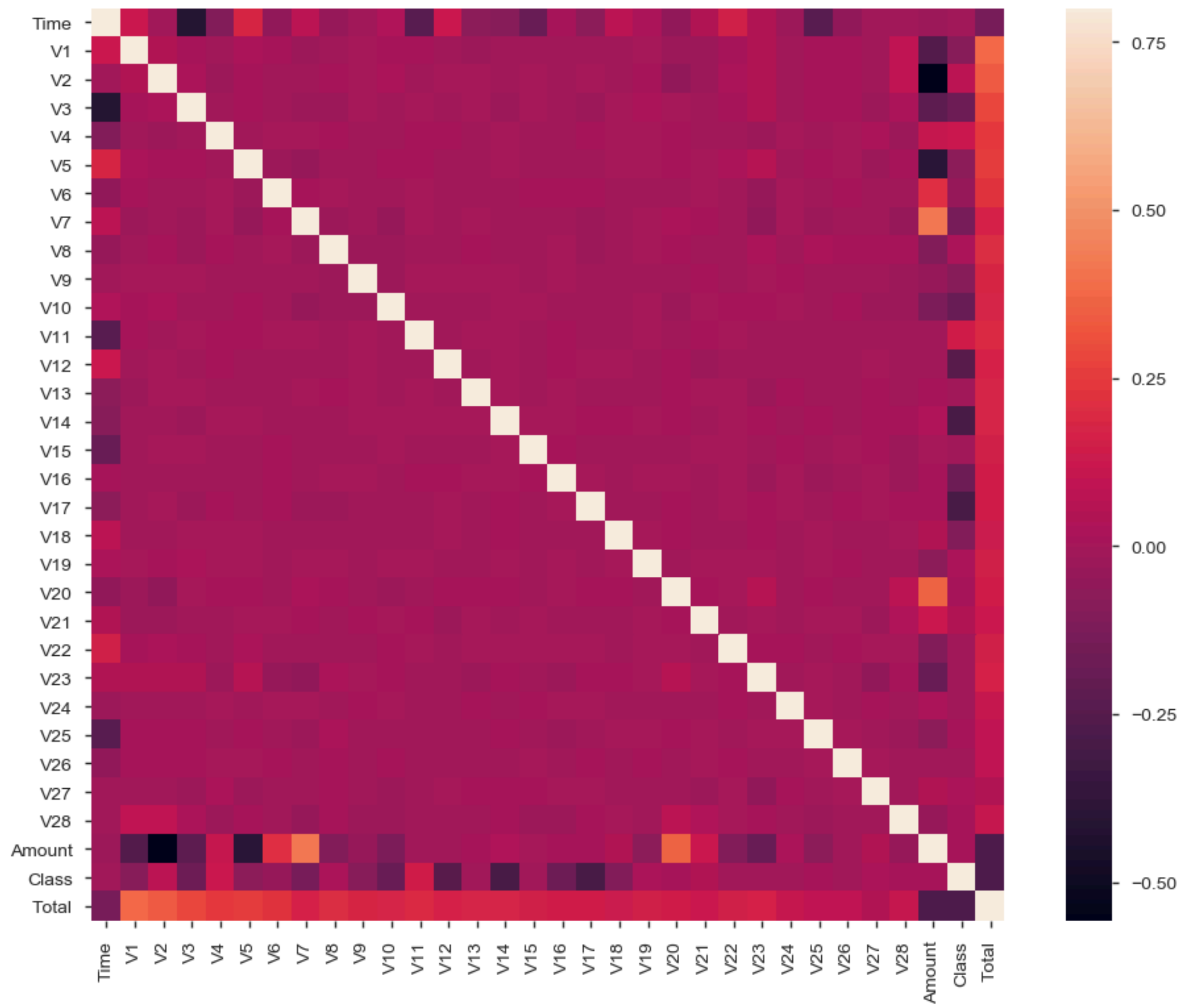


Because the row totals did not present useful data, I also looked at each V column category to determine what segments would more easily separate potential fraudulent transactions for the legitimate transactions.

Feature Density Plot



Correlatation Matrix Heatmap



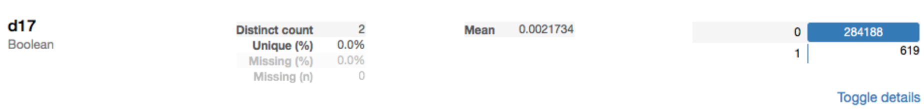
I then wrote code to attempt to isolate the fraudulent transactions using for loops.

```
for i in range(len(wip)):
    if (wip['V1'][i] < -3):
        D01.append(1)
    else:
        D01.append(0)

for i in range(len(wip)):
    if (wip['V2'][i] > 2):
        D02.append(1)
    else:
        D02.append(0)

for i in range(len(wip)):
    if (wip['V3'][i] < -3):
        D03.append(1)
    else:
        D03.append(0)

for i in range(len(wip)):
    if (wip['V4'][i] > 2):
        D04.append(1)
    else:
        D04.append(0)
```



V17 appeared to have the best promise with no tweaking of the code. However, this isn't machine learning techniques so this method was scrapped.

A straight logistic regression appears to give promising results on the surface.

```
logisticRegr = LogisticRegression()
logisticRegr.fit(X_train, y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

```
predictions = logisticRegr.predict(X_test)
```

```
score = logisticRegr.score(X_train, y_train)
print(score)
```

```
0.999095876583
```

```
score = logisticRegr.score(X_test, y_test)
print(score)
```

```
0.998911555072
```

```
cm = metrics.confusion_matrix(y_test, predictions)
print (cm)
```

```
[[56851    13]
 [   49    49]]
```

```
predictions = logisticRegr.predict(X_train)
cm = metrics.confusion_matrix(y_train, predictions)
print (cm)
```

```
[[227410     41]
 [   165    229]]
```

The scores look good. However, when running the confusion matrix and comparing both legitimate transactions and fraudulent transactions, there is an issue. Legitimate transactions have great results. However, fraudulent transactions at best have a 50% score. This number is totally unacceptable.

```
Isolation Forest: 75
0.997366665496
             precision    recall  f1-score   support

          0       1.00      1.00      1.00     28432
          1       0.24      0.24      0.24        49

avg / total       1.00      1.00      1.00     28481
```

```
1 confusion_matrix=pd.crosstab(y_test,pred,rownames=['Actual'],colnames=['prediction'])
2 confusion_matrix
```

prediction	0	1
Actual		
0	5642	46
1	1	8

```
1 report=classification_report(y_test,pred)
2 print(report)
```

```
             precision    recall  f1-score   support

          0       1.00      0.99      1.00      5688
          1       0.15      0.89      0.25         9

avg / total       1.00      0.99      0.99      5697
```

```
model
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
confusion matrix
[[8532    1]
 [   5    7]]
classification report
             precision    recall  f1-score   support

          0       1.00      1.00      1.00      8533
          1       0.88      0.58      0.70        12

avg / total       1.00      1.00      1.00      8545

Accuracy : 0.999298
Area under the curve : 0.791608
```

Isolation Forest and Random Forest classifiers/models fair even worse.

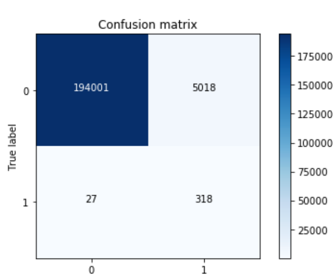
This is because the dataset has over 99% legitimate transactions and these models are unable to analyze unbalanced datasets properly.

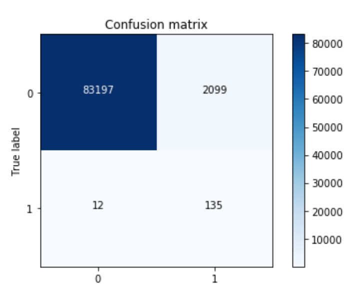
THE SOLUTION:

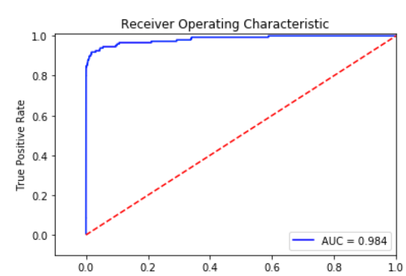
The learning phase and the subsequent prediction of machine learning algorithms can be affected by the problem of unbalanced datasets. The balancing issue corresponds to the difference of the number of samples in the different classes. The decision function of the linear SVM is highly impacted. With a greater unbalanced ratio, the decision function favor the class with the larger number of samples, usually referred as the majority class.

One way to fight this issue is to generate new samples in the classes which are under-represented. The most naive strategy is to generate new samples by randomly sampling with replacement the current available samples. The RandomOverSampler offers such scheme

Apart from the random sampling with replacement, there are two popular methods to over-sample minority classes: (i) the Synthetic Minority Oversampling Technique (SMOTE) [CBHK2002] and (ii) the Adaptive Synthetic (ADASYN) [HBGL2008] sampling method. The solution for this dataset was to use the SMOTE to resample the data and logistic regression to perform the analysis with these results:







APPENDIX – V Numbers compared

Fruadalent Transactions

Distinct count	473	Mean	-4.7719
Unique (%)	96.1%	Minimum	-30.552
Missing (%)	0.0%	Maximum	2.1324
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V1
Numeric

Distinct count	473	Mean	3.6238
Unique (%)	96.1%	Minimum	-8.4022
Missing (%)	0.0%	Maximum	22.058
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V2
Numeric

This variable is highly correlated with *V1 and should be ignored for analysis*

Correlation	0.90788

V3
Numeric

Distinct count	473	Mean	4.542
Unique (%)	96.1%	Minimum	-1.3133
Missing (%)	0.0%	Maximum	12.115
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V4
Numeric

Distinct count	473	Mean	-3.1512
Unique (%)	96.1%	Minimum	-22.106
Missing (%)	0.0%	Maximum	11.095
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V5
Numeric

Distinct count	473	Mean	-1.3977
Unique (%)	96.1%	Minimum	-6.4063
Missing (%)	0.0%	Maximum	6.4741
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V6
Numeric

Distinct count	473	Mean	-5.5687
Unique (%)	96.1%	Minimum	-43.557
Missing (%)	0.0%	Maximum	5.8025
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V7
Numeric

Distinct count	473	Mean	0.57064
Unique (%)	96.1%	Minimum	-41.044
Missing (%)	0.0%	Maximum	20.007
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V8
Numeric

Distinct count	473	Mean	-2.5811
Unique (%)	96.1%	Minimum	-13.434
Missing (%)	0.0%	Maximum	3.3535
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V9
Numeric

Legitimate Transactions

Distinct count	275190	Mean	0.0082577
Unique (%)	96.8%	Minimum	-56.408
Missing (%)	0.0%	Maximum	2.4549
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	-0.0062709
Unique (%)	96.8%	Minimum	-72.716
Missing (%)	0.0%	Maximum	18.902
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.012171
Unique (%)	96.8%	Minimum	-48.326
Missing (%)	0.0%	Maximum	9.3826
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	-0.0078599
Unique (%)	96.8%	Minimum	-5.6832
Missing (%)	0.0%	Maximum	16.875
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.0054531
Unique (%)	96.8%	Minimum	-113.74
Missing (%)	0.0%	Maximum	34.802
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.0024187
Unique (%)	96.8%	Minimum	-26.161
Missing (%)	0.0%	Maximum	73.302
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.0096365
Unique (%)	96.8%	Minimum	-31.765
Missing (%)	0.0%	Maximum	120.59
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	-0.00098747
Unique (%)	96.8%	Minimum	-73.217
Missing (%)	0.0%	Maximum	18.709
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.0044666
Unique (%)	96.8%	Minimum	-6.2907
Missing (%)	0.0%	Maximum	15.595
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Fruadalent Transactions

V10 Numeric

Distinct count	473	Mean	-5.6769
Unique (%)	96.1%	Minimum	-24.588
Missing (%)	0.0%	Maximum	4.0314
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V11 Numeric

Distinct count	473	Mean	3.8002
Unique (%)	96.1%	Minimum	-1.7022
Missing (%)	0.0%	Maximum	12.019
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	[illegible]		

V12 Numeric

Distinct count	473	Mean	-6.2594
Unique (%)	96.1%	Minimum	-18.684
Missing (%)	0.0%	Maximum	1.3759
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V13 Numeric

Distinct count	473	Mean	-0.10933
Unique (%)	96.1%	Minimum	-3.1278
Missing (%)	0.0%	Maximum	2.8154
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V14 Numeric

Distinct count	473	Mean	-6.9717
Unique (%)	96.1%	Minimum	-19.214
Missing (%)	0.0%	Maximum	3.4424
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V15 Numeric

Distinct count	473	Mean	-0.092929
Unique (%)	96.1%	Minimum	-4.4989
Missing (%)	0.0%	Maximum	2.4714
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V16 Numeric

Distinct count	473	Mean	-4.1399
Unique (%)	96.1%	Minimum	-14.13
Missing (%)	0.0%	Maximum	3.1397
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V17 Numeric

This variable is highly correlated with V16 and should be ignored for analysis Correlation 0.96015

V18 Numeric

This variable is highly correlated with V17 and should be ignored for analysis Correlation 0.97149

V19 Numeric

Distinct count	473	Mean	0.68066
Unique (%)	96.1%	Minimum	-3.6819
Missing (%)	0.0%	Maximum	5.2283
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Legitimate Transactions

Distinct count	275190	Mean	0.0098237
Unique (%)	96.8%	Minimum	-14.741
Missing (%)	0.0%	Maximum	23.745
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	[illegible]		

Distinct count	275190	Mean	-0.0065761
Unique (%)	96.8%	Minimum	-4.7975
Missing (%)	0.0%	Maximum	10.002
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.010832
Unique (%)	96.8%	Minimum	-15.145
Missing (%)	0.0%	Maximum	7.8484
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.0001892
Unique (%)	96.8%	Minimum	-5.7919
Missing (%)	0.0%	Maximum	7.1269
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.012064
Unique (%)	96.8%	Minimum	-18.392
Missing (%)	0.0%	Maximum	10.527
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.00016081
Unique (%)	96.8%	Minimum	-4.3913
Missing (%)	0.0%	Maximum	8.8777
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.0071641
Unique (%)	96.8%	Minimum	-10.116
Missing (%)	0.0%	Maximum	17.315
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.011535
Unique (%)	96.8%	Minimum	-17.098
Missing (%)	0.0%	Maximum	9.2535
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	[illegible]		

Distinct count	275190	Mean	0.0038872
Unique (%)	96.8%	Minimum	-5.3667
Missing (%)	0.0%	Maximum	5.0411
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	[illegible]		

Distinct count	275190	Mean	-0.0011779
Unique (%)	96.8%	Minimum	-7.2135
Missing (%)	0.0%	Maximum	5.592
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Fruadalent Transactions

V20 Numeric

Distinct count	473	Mean	0.37232
Unique (%)	96.1%	Minimum	-4.1282
Missing (%)	0.0%	Maximum	11.059
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V21 Numeric

Distinct count	473	Mean	0.71359
Unique (%)	96.1%	Minimum	-22.798
Missing (%)	0.0%	Maximum	27.203
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V22 Numeric

Distinct count	473	Mean	0.014049
Unique (%)	96.1%	Minimum	-8.887
Missing (%)	0.0%	Maximum	8.362
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V23 Numeric

Distinct count	473	Mean	-0.040308
Unique (%)	96.1%	Minimum	-19.254
Missing (%)	0.0%	Maximum	5.4662
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V24 Numeric

Distinct count	473	Mean	-0.10513
Unique (%)	96.1%	Minimum	-2.028
Missing (%)	0.0%	Maximum	1.0914
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V25 Numeric

Distinct count	473	Mean	0.041449
Unique (%)	96.1%	Minimum	-4.7816
Missing (%)	0.0%	Maximum	2.2082
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V26 Numeric

Distinct count	473	Mean	0.051648
Unique (%)	96.1%	Minimum	-1.1527
Missing (%)	0.0%	Maximum	2.7453
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V27 Numeric

Distinct count	473	Mean	0.17057
Unique (%)	96.1%	Minimum	-7.2635
Missing (%)	0.0%	Maximum	3.0524
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

V28 Numeric

Distinct count	473	Mean	0.075667
Unique (%)	96.1%	Minimum	-1.8693
Missing (%)	0.0%	Maximum	1.7794
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Legitimate Transactions

Distinct count	275190	Mean	-0.00064429
Unique (%)	96.8%	Minimum	-54.498
Missing (%)	0.0%	Maximum	39.421
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		

Distinct count	275190	Mean	-0.0012348
Unique (%)	96.8%	Minimum	-34.83
Missing (%)	0.0%	Maximum	22.615
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	-2.4311e-05
Unique (%)	96.8%	Minimum	-10.933
Missing (%)	0.0%	Maximum	10.503
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	6.9752e-05
Unique (%)	96.8%	Minimum	-44.808
Missing (%)	0.0%	Maximum	22.528
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	0.00018193
Unique (%)	96.8%	Minimum	-2.8366
Missing (%)	0.0%	Maximum	4.5845
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	-7.1726e-05
Unique (%)	96.8%	Minimum	-10.295
Missing (%)	0.0%	Maximum	7.5196
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	-8.9376e-05
Unique (%)	96.8%	Minimum	-2.6046
Missing (%)	0.0%	Maximum	3.5173
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	-0.00029518
Unique (%)	96.8%	Minimum	-22.566
Missing (%)	0.0%	Maximum	31.612
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

Distinct count	275190	Mean	-0.00013094
Unique (%)	96.8%	Minimum	-15.43
Missing (%)	0.0%	Maximum	33.848
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		

APPENDIX – Kaggle Details about the dataset.
Context
It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

Content
The datasets contains transactions made by credit cards in September 2013 by european cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Inspiration
Identify fraudulent credit card transactions.

Given the class imbalance ratio, we recommend measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC). Confusion matrix accuracy is not meaningful for unbalanced classification.

Acknowledgements
The dataset has been collected and analysed during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection. More details on current and past projects on related topics are available on http://mlg.ulb.ac.be/BruFence and http://mlg.ulb.ac.be/ARTML

Please cite: Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015