Data Exploration and cleaning

Gus Vu

2023-05-22

# The Data

The dataset 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-dependant cost-sensitive learning. Feature ‘Class’ is the response variable and it takes value 1 in case of fraud and 0 otherwise.

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.

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 <https://www.researchgate.net/project/Fraud-detection-5> and the page of the DefeatFraud project

# Data Exploration

dim(creditcard)

## [1] 284807 31

head(creditcard, 6)

## Time V1 V2 V3 V4 V5 V6  
## 1 0 -1.3598071 -0.07278117 2.5363467 1.3781552 -0.33832077 0.46238778  
## 2 0 1.1918571 0.26615071 0.1664801 0.4481541 0.06001765 -0.08236081  
## 3 1 -1.3583541 -1.34016307 1.7732093 0.3797796 -0.50319813 1.80049938  
## 4 1 -0.9662717 -0.18522601 1.7929933 -0.8632913 -0.01030888 1.24720317  
## 5 2 -1.1582331 0.87773675 1.5487178 0.4030339 -0.40719338 0.09592146  
## 6 2 -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

We can see that our data set has 31 observations. The data is already broken down into principal component analysis. This is to protect identities of the observations. But the 3 variables that are not broken down are time, amount, and class.

From kaggle: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

We know that time is the number of seconds elapsed between this transaction and the first transaction in the dataset. Amount is the transaction amount. Class is 1 for fraud, and 0 otherwise. v1-v28 may be result of a PCA Dimensionality reduction to protect user identities and sensitive features(v1-v28).

table(creditcard$Class)

##   
## 0 1   
## 284315 492

summary(creditcard$Amount)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 5.60 22.00 88.35 77.17 25691.16

var(creditcard$Amount)

## [1] 62560.07

We observe that only 492 observations are fraud. The average transaction amount is 88.35, while the median is 22.0. This may mean that our data is skewed or not normaly distributed. We also notice that transaction amount has a large variance 62560.07.

# Data Manipulation

Now that I have explored the data. I am going to scale the amount variable. This is also known as standardizing the data. Instead of measuring individual data points we measure how many standard deviations they are away from the center. We do this so that we do not have any extreme points that may mess up our model.

creditcard$Amount=scale(creditcard$Amount)  
NewData=creditcard[,-c(1)]  
head(NewData)

## 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 0.24496383 0  
## 2 0.1671704 0.1258945 -0.008983099 0.01472417 -0.34247394 0  
## 3 -0.3276418 -0.1390966 -0.055352794 -0.05975184 1.16068389 0  
## 4 0.6473760 -0.2219288 0.062722849 0.06145763 0.14053401 0  
## 5 -0.2060096 0.5022922 0.219422230 0.21515315 -0.07340321 0  
## 6 -0.2327938 0.1059148 0.253844225 0.08108026 -0.33855582 0