University of Jean Monnet

Data Mining Lab Project

PORTO SEGURO SAFE DRIVER PREDICTION (Kaggle Competition)

Author: Maedeh Afshari *Student Number: 16008290*

1 Problem Understanding

The Porto Seguro's Safe Driver Prediction competition hosted by Kaggle, one of Brazil's largest auto and homeowner insurance companies. Inaccuracies in car insurance company's claim predictions raise the cost of insurance for good drivers and reduce the price for bad ones. In this competition, the challenged is to build a model that predicts the probability that a driver will initiate an auto insurance claim in the next year.

2 Data Understanding

Here is an excerpt of the the data description for the competition:

Features that belong to similar groupings are tagged as such in the feature names (e.g., ind, reg, car, calc). Feature names include the postfix bin to indicate binary features and cat to indicate categorical features.
Features without these designations are either continuous or ordinal. Values of -1 indicate that the feature was missing from the observation. The target columns signifies whether or not a claim was filed for that policy holder [1].

The train set has **595,212** labeled observation for each costumer and **59** features. In this work a part of data set has been choosen (n = 100,000).The labels indicate whether the customer filed an insurance claim or not. About **3,671** examples have label 1, and the remaining **96,329** have label 0 (Figure 1). There is no duplicated rows in this data set.

Figure 1: Distribution of the target

2.1 Correlation between Features

Continues features like ind,car,reg,calc is separated from other features binary and categorical and save them as **cont_vars**. Then the correlation between them checked (Figure 2).

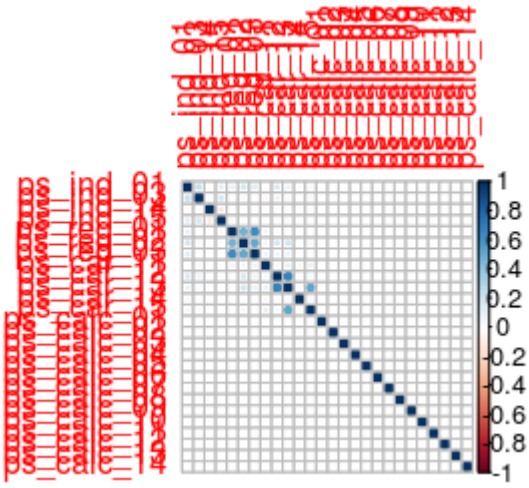

Figure 2: Correlation Matrix of Continuous Features

In figure 2, positive correlations are displayed in blue and negative correlations in red color. The darker blue shows more correlation. We can observe that there is a relation between ind, reg and car features and no relation between calc features. So it's better to break the correlations down by feature groups (Figure 3, Figure 4, Figure 5).

2.2 Missing Values

Some values in training set are missing and indicated by -1. To find put which variables have more missing values than others, -1 converted to NA (Figure 6) and about 141891 missing value in train set.
As the figure 6 shows, **ps_car_03_cat** has around 70 %, **ps_car_05_cat** has around 45 %, **ps_reg_03** has around 18 % and the remaining three have only a small percent missing data.

Missing value in related columns computed by median.

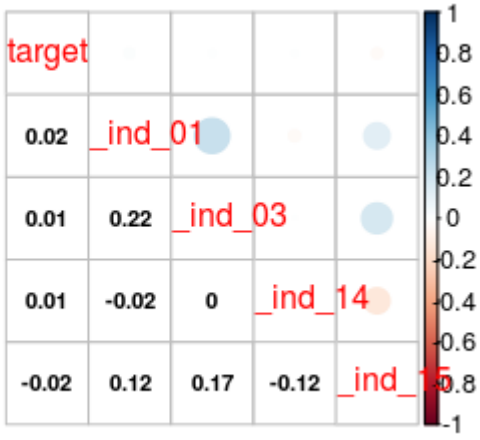

Figure 3: Correlation of _ind_ Features

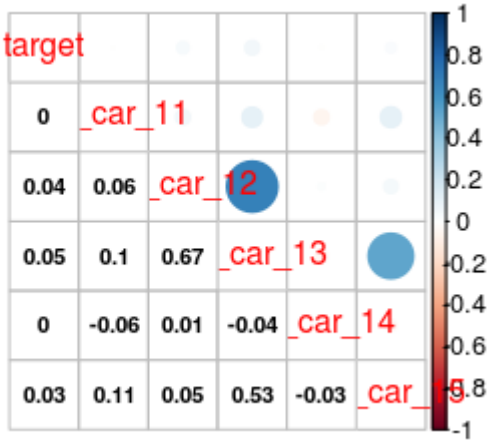

Figure 4: Correlation of _car_ Features

3 Data Preparation

3.1 PCA

As PCA can be applied only on numerical data so categorical data must be converted to numerical data. Also for missing values, they imputed by median in each column with missing values.[2]
As we have a data set of dimension **595212(n)** $*$ **59p**, *min(n-1, p)* principal component can be constructed.

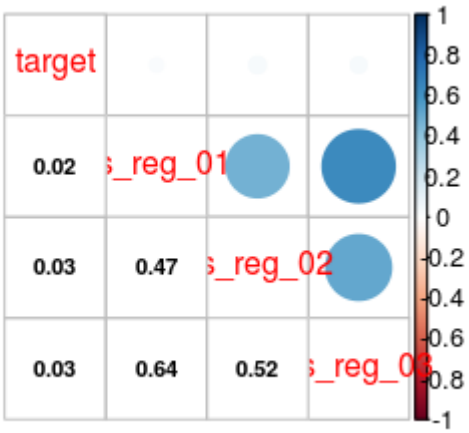

Figure 5: Correlation of _reg_ Features

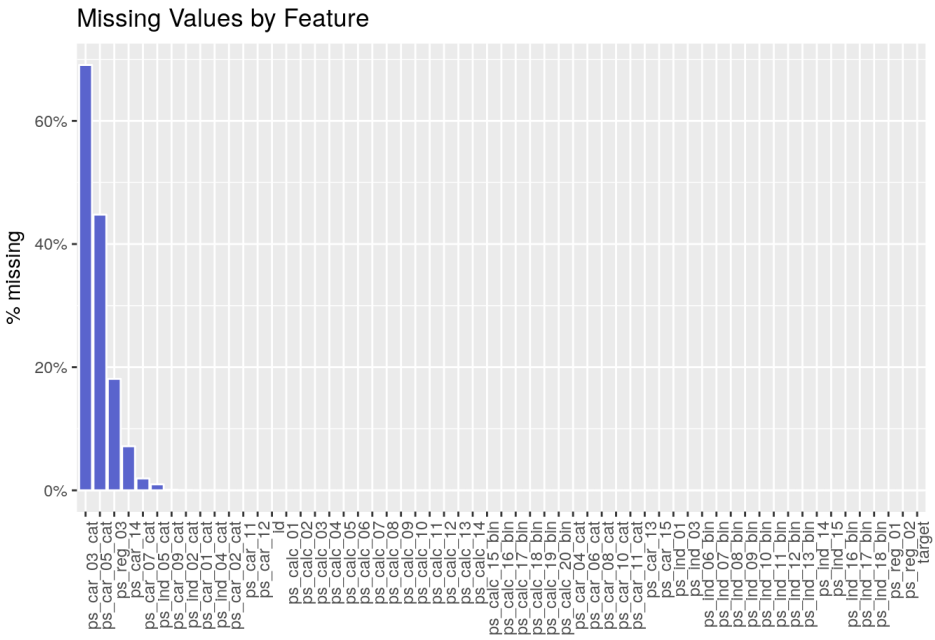

Figure 6: Percent of missing values by Features

The base R function **prcomp()** is used to perform PCA. By default, it centers the variable to have mean equals to zero. With parameter scale. = T, we normalize the variables to have standard deviation equals to 1.

The **prcomp()** function results in 5 useful measures: "sdev" "rotation" "center" "scale" "x".

Center and scale refers to respective mean and standard deviation of the variables that are used for normalization prior to implementing PCA.

The rotation measure provides the principal component loading. Each column of rotation matrix contains the principal component loading vector. This is the most important measure we should be interested in.
Let's look at first 4 principal components and first 5 rows:

	PC1	PC2	PC3	PC4
ps_ind_01	0.13563402	-0.129230725	0.183955318	0.20262375
ps_ind_02_cat	-0.02271765	-0.021945770	0.172115595	-0.14076546
p_ind_03	0.04681928	-0.053968952	0.066818072	0.20167924
ps_ind_04_cat	0.04157601	-0.019690754	0.146940398	-0.25751189
ps_ind_05_cat	-0.01346674	-0.004847732	-0.001104821	0.01600981

Figure 7 shows the resultant principal components (Figure 7):

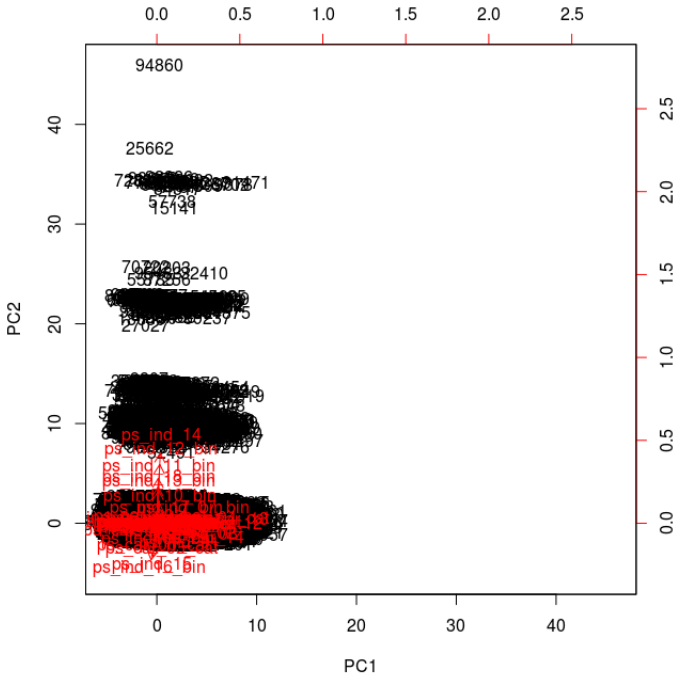

Figure 7: BiPlot

We aim to find the components which explain the maximum variance. This is because, we want to retain as much information as possible using these components. So, higher is the explained variance, higher will be the information contained in those components.check variance of first 10 components (Figure 8):

```
> #check variance of first 10 components
> pr_var[1:10]
 [1] 3.771136 2.680510 2.253407 1.931466 1.815689 1.748437 1.485871 1.402329 1.229464 1.200089
```

Figure 8: variance of first 10 components

To compute the proportion of variance explained by each component, we simply divide the variance by sum of total variance. This results in (Figure 9):

```
> prop_varex[1:20]
 [1] 0.06616028 0.04702649 0.03953346 0.03388537 0.03185420 0.03067433 0.02606792 0.02460226
 [9] 0.02156955 0.02105420 0.01938268 0.01856637 0.01832692 0.01798829 0.01786512 0.01784153
[17] 0.01781698 0.01773818 0.01772722 0.01769354
```

Figure 9: variance of first 10 components

This shows that first principal component explains 6,6% variance. Second component explains 4.7% variance. Third component explains 3.9% variance and so on. So, to decide how many components should we select for modeling part, scree plot is used. A scree plot is used to access components or factors which explains the most of **variability** in the data (Figure 10 , Figure 11).

So the first 20 pca are selected to make prediction model.

4 Modeling

4.1 Logistic Regression

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.
As we are going to predict the binary outcome, the Logistic regression is chosen by first 20 pca's and the result has shown in (figure 12).

5 Conclusion

This work attempted to conduct dimensionality reduction on our dataset by using PCA. But The result showed that most of the variance in the training set

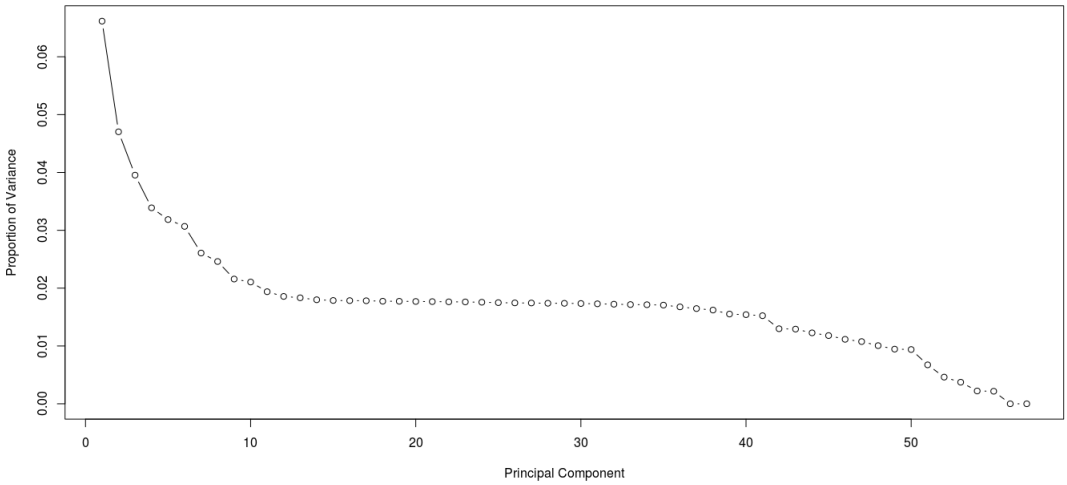

Figure 10: screePlot

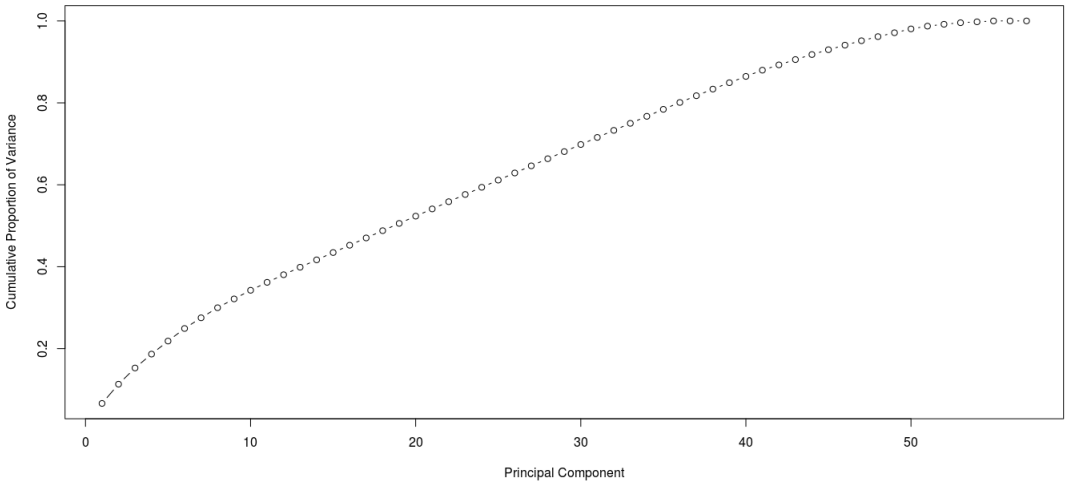

Figure 11: cumulative scree plot

can be explained by a single principal component. This suggests the dataset is incompatible with PCA, which may be due to the large number of categorical and binary features it contains.

id	target
0	0.02581691
1	0.03169096
2	0.02896749
3	0.01927865
4	0.02835009
5	0.03102363
6	0.03751411
8	0.02847503
10	0.04526372
11	0.05844073
12	0.03016546
14	0.03065697
15	0.06565075
18	0.04695451
21	0.04240905
23	0.02978239
24	0.03526370
25	0.02532265
27	0.01578659
29	0.04366598
30	0.03889624
31	0.04700266
32	0.05028059
33	0.01577077
37	0.03220469
38	0.02752895
39	0.07226840
40	0.03344879
41	0.03442723
42	0.02137262
44	0.03377601

Figure 12: Prediction

References

[1] Porto seguro's safe driver prediction. https://www.kaggle.com/c/porto-seguro-safe-driver-prediction. Accessed: 2018-03-12.

[2] practical-guide-principal-component-analysis-python. https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/. Accessed: 2018-03-12.