Capstone Project-2

Title- Porto Seguro Safe Driver Prediction

**Problem-**

Inaccuracies in car insurance company's claim predictions raise the cost of insurance for good drivers and reduce the price for bad ones. If a driver has claimed the insurance, then there is a possibility that the drivers share some common attributes with another driver who have filed the claim. It is like the bad drivers will have the same feature set and the good ones will share the same feature set. It would be great to create a model which can classify between the two drivers group and adjust the insurance price accordingly.

**The Client**-

The Clients are auto insurance company which can use the prediction models to classify among the good and bad drivers. Also, the auto insurance company can issue guidelines on how to become a good driver. Hence drivers will have a chance to improve their records and lower their insurance premium. The first client will be Porto Seguro, Brazil’s largest auto insurance company and the donor of this data set.

**Data-**

The Data is available on Kaggle. Here is the link-

https://www.kaggle.com/c/porto-seguro-safe-driver-prediction

The data is provided by the Porto Seguro. The data is highly anonymized, and no information is provided about the features. There are almost 54 features for a single driver. The data is divided into the Train and test. The train data has a target column which has 0 for no claim and 1 for the claim. Hence, it is a binary classification problem.

The total number of features is 57.

Total categorical features are 14.

Total binary features are 17.

Total continuous features are 26.

**Approach and Results** –

With the continuous features, we can do a correlation analysis and also do a PCA to see which features has the most impact on the final result. The results are shown in the later section of this report.

Binary Variable Analysis-

The results of the binary variable analysis are –

1) 'ps\_ind\_08\_bin' variable plays a crucial role on the target variable, 0ut of 3.5 %, 2.9% did not files a claim when 'ps\_ind\_08\_bin' is false and 0.6% filed claim when it is true. Hence a good variable to learn.

2) Same for the variables, 'ps\_ind\_10\_bin', 'ps\_ind\_11\_bin', 'ps\_ind\_12\_bin', 'ps\_ind\_13\_bin'. Aslo, 'ps\_calc\_15\_bin' and 'ps\_ind\_20\_bin' have a strong association with the target variable.

3) String association means that how the target variable switches between 0 and 1 with respect to the binary input.

4) Please note that since we have binary variable on both sides, feature space and target space, it is irrelevant to talk about the correlation, rather we can try to estimate how strong an input variable is associated with the output.

Categorical Variable Analysis-

There are two assumptions for the categorical variables-1) The categorical variable is independent; it means that the variables are something like car categories and are independent to each other and target as well. 2) The categorical variables are dependent and correlated with the target. It means that the variable presents something like the citations/tickets from the cops. Some customers might have 4 -5 tickets/citation for reckless driving or any other reason and those customers are kept in the high-risk area. Customers with 2-3 tickets are kept in the moderate area and customers with 0 to 1 tickets are kept in the safe zone.

If the second assumption is true, then there is supposed to be a strong correlation with the target variable. There are a total of 14 categorical features. The best way to check a correlation among categorical and binary variable is to conduct a polychoric correlation. However, there is no library or package which estimates polychoric, hence this idea is dropped.

To check if our second assumption is correct, we can assume the categorical to be continuous variable and use point biserial correlation coefficient from scipy package, which calculates correlation between the binary and continuous variables. This will solve the purpose.

After running a correlation analysis, we found that our first assumption is correct and there is no support the second assumption.

Pearson Correlation Analysis-

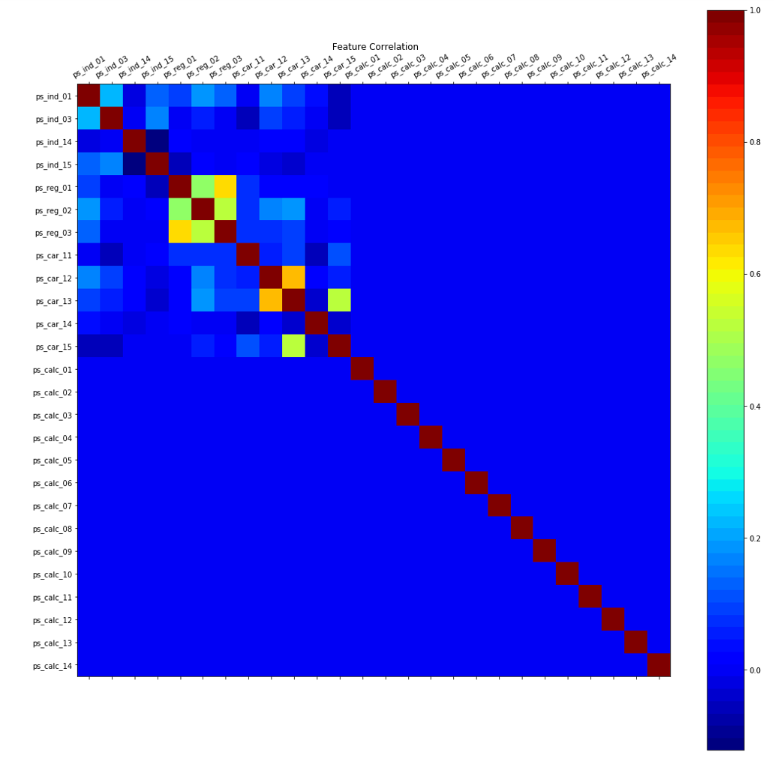
From the below matrix it is very clear that the following variables have a strong correlation-

1) 'ps\_car\_15' with 'ps\_car\_13', a strong positive correlation is found.

2) 'ps\_car\_13' with 'ps\_car\_12', a strong positive correlation is found.

3) 'ps\_reg\_03' with 'ps\_reg\_01' and 'ps\_reg\_02', a strong positive correlation is found.

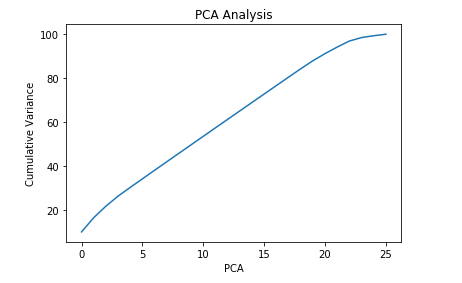
The data is anonymized, so it is hard to determine what does each of these variable means. For example, variable 'ps\_reg\_03', might have to do something with the registering of the user in a certain region in Brazil. PS must have divided the Brazil into different regions, and these variables are the indicators.



PCA (Principal Component Analysis)

PCA is conducted to reduce the dimensionality of the dataset.

We found that with 24 PC are good enough to capture all the variance in the data. Below is a graph showing the PCA variance.



After 24 PCA there is not much difference in variance. But if we choose 24, we are still not able to reduce the dimensionality of data at large. So we plan to stick with the 5 PC only.

Downsampling-

Our training data is highly imbalanced with only 3% of user data who has claimed an insurance. The approach is to undersample the 'No Claim' class and brings the sample numbers equivalent to the 'Claim' class.

Tomek links are used to downsample the data. Tomek Links create clusters of classes representing different data points, Now Tomek links are defined as the links which exist if the two samples are the nearest neighbors of each other. Hence, training data is undersampled and represent unique samples. We will try running the algorithm and see how much it under samples.

After running the Tomek Links algorithm, only 6501 samples were removed. To balance the classes random sampling techniques are used to take 1:3 ratio of classes.

Continuous Variables

The strategy is to divide the data into two groups. One represents the positive group, and the other one represents the negative group (Class A and Class B). We analyzed the continuous variables for both the groups and found that behavior of the variable in the groups is identical.

We calculated the mean, standard deviation and the distribution curves of variables from both the groups.

1) Our analysis indicates that there is no significant difference between the continuous variables in these two groups.

2) It will be difficult for the ML models to learn about them because there is no variance among these variables in these two different subsets of data.

IPython notebook link- <https://github.com/ayusmittal/SpringBoard-Projects/blob/master/CapStone-Project-2/Milestone.ipynb>

. . .

**Machine Learning**

The following Machine Learning models are implemented-

1. Random Forest
2. Ensemble Learning (Using Decision Tree as base learner)
3. Deep Neural Network

None of them gave a good accuracy on our problem statement. Several feature engineering techniques were applied from the knowledge gained during the EDA. Our big concern is to choose the correct ‘Claim’ class.

Here are the results summarized from different simulations and different feature set selection-

|  |  |  |
| --- | --- | --- |
|  | Random Forest |  |
| Feature Engineering | % accuracy | AUC |
| All Features | 96.30% | 0.54 |
| Only Binary features | 96.30% | 0.56 |
|  | Under-Sampling |  |
| All Features | 72.18% | 0.62 |
| Binary + Continuous | 72.18% | 0.6 |
| Only Binary features | 72.05% | 0.57 |
|  | PCA in Random Forest |  |
| 5 PCA + Binary | 71.80% | 0.59 |
| 2 PCA + + Binary | 76.60% | 0.55 |
| 10 PCA + + Binary | 71.90% | 0.6 |
| 1 PCA + Binary | 61.35% | 0.53 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Ensemble Learning(depth =40, n= 400) |  |  |  |
| Feature Engineering | % accuracy | AUC |  |  |
| Binary | 72.04% | 0.56 |  |  |
| Selecting 200 Features using chi^2 test | 69.80% | 0.58 |  |  |

|  |  |  |
| --- | --- | --- |
|  | Deep Learning |  |
| Feature Engineering | % Accuracy | AUC |
| All Features | 63.60% | 0.52 |
| Binary features | 72.08% | 0.57 |
| 5 PCA + Binary | 69.12% | 0.52 |
| Selecting 10 Features using chi^2 test | 72.20% | 0.61 |
| Selecting 30 Features using chi^2 test | 71.60% | 0.57 |
| Selecting 500 Features using chi^2 test | 70.80% | 0.58 |
| Selecting 100 Features using chi^2 test | 70.07% | 0.55 |
| Selecting 150 Features using chi^2 test | 69.60% | 0.58 |
| Selecting 200 Features using chi^2 test | 68.10% | 0.55 |

Please note- I have included accuracy in the above tables, but it is not relevant to our problem. Because of the class imbalance, Area under the curve of ROC for the minority class (‘Claim Class') is of our interest, and I have calculated AUC for each classifier.

Also, the architecture used for the Deep neural network is a six-layer network with four hidden layers, the hidden layer is 256 -> 128 -> 64 -> 32. Using a sigmoid function in the output layer to predict the probability of classes.

The Chi^2 feature selection process comes from the Univariate selection process; the process is briefly described below.

Feature Selection-

Scikit learns several feature selection techniques inbuilt in it. Three different techniques used are –

1. Recursive feature elimination
2. Feature importance
3. Univariate Selection.

Recursive Feature Selection-

The Recursive Feature Elimination (or RFE) works by recursively removing attributes and building a model on those attributes that remain.

It uses the model accuracy to identify which attributes (and the combination of attributes) contribute the most to predicting the target attribute. Any model can be used; I tried using a random forest. Unfortunately, while running this, the optimum solution was never reached. The system was in an infinite loop trying to reduce the features and never succeeded in doing any. This idea was dropped.

Feature Importance-

Bagged decision trees like Random Forest and Extra Trees can be used to estimate the importance of features. The coefficients were checked for this method, and all of them were very small. It was hard to decide that which feature to pick since the coefficients associated with them all are very small. This idea is also dropped.

Univariate Selection-

Statistical tests can be used to select those features that have the strongest relationship with the output variable. Chi-squared (chi^2) statistical test for non-negative features is used to select some of the top features from the entire data set. The top features selected are 10, 30, 50 and 100.

**Final Conclusion-**

1. DataSet is a mixture of the categorical, binary and the continuous variables.
2. The target class (binary) is highly imbalanced, with only 3% of samples belonging to the 'CLAIM' class.
3. The binary variable analysis is conducted to see how strongly each variable is associated with the target variable. We found that the binary variables are the most important in predictions. For example, these binary variables might be questions like ‘Have you ever met with an accident before.'
4. The continuous variables (calc and values) does not represent any variance in two groups of data (Claim Vs. No Claim). This is concluded by looking at the Mean, distribution curves and Standard deviation of different samples.
5. Pearson correlation is conducted to see the relationships among the continuous variables. Which comes out be very low again except for a few.
6. PCA is done and concluded that 24 PC represents the 100 % of the variance, but for the sake of simplicity, we will keep PC to be four only. Hence, reducing the dimensionality.
7. We also found that our assumption regarding the categorical variables is wrong.
8. Finally, we concluded that there is no variance in the continuous variable of the different group of people who claimed and who do not claim the insurance.
9. In feature selection process the univariate feature selection process is implemented and used. It helped in reducing the feature set to 10,20, 50 and 100.
10. The best performing model for this problem set is Deep Learning and Random Forest.
11. Deep Learning models are prone to overfitting most of the time, even with the higher number of epochs, the accuracy did not improve.
12. At last, to solve the problem in great detail, PS should at least reveal the feature name, and what they mean it will be a great help in understanding and solving problem.