Machine Learning Engineer Nanodegree

Capstone Proposal

Paulo Roberto de Oliveira Castro October 12th, 2017

Proposal

Domain Background

Insurance is a data-driven business. Since it's beggining, it's possible to observe[1] that the area was always focused on gathering information about a sutuation that involves risk (driving a car, having a house, etc.) and then measuring that risk, so it's possible to profit by assuming someone's risk and getting payed for that in exchange.

However, especially in Brasil, this is a rather expensive business for the final costumer. Buying new car, even if you're a very good driver, means a huge insurance spending. Sometimes it even prevents the costumer from buying it at all. Therefore, a good prediction about what is the real risk for a specific client means that the provider can charge a more reasonable amount of money for their insurance products. Machine Learning is therefore, one of the ways to achieve this[2]

Problem Statement

Porto Seguro is one of the biggest insurance company in Brasil. Currently, they are looking for new modelling techiniques and algorithms to improve their prediction as whether the policy holder (their client) will initiate an auto insurance claim or not. This problem could be replicated for every insurance company, specially in Brasil, and the performance can be quantified by any of the means that any binary classification problem can be.

Datasets and Inputs

For this specific problem, Porto Seguro[2] provided a dataset with dozens of features for more than five hundred thousand clients, alongside with a flag if that client claimed of not. Altough this is a static dataset, in the sense that there is no time series of their clients events that we could use for determining the time of the claim, it's more than enough to get good results prediction the target. This dataset is available for educational purposes and it's easily obtained using the Kaggle platform[4].

The features are anonymized, in the sense we don't know the meaning of any of them, but they contain cadastral information and the client's behavior as a insurance client, and range from continuous to categorical features. As a total, this dataset contains 595212 examples with 57 features along an indication whether a claim was made or not by that client (a binary value). The target is also highly unbalanced (only 3.7% of the clients have claimed).

Solution Statement

Any techiniques for classification, in special binary classification can be done to solve this problem. As an example, logisitic regression, boosting methods, nayve bayes, k-nearest-neighbors, and all the other algorithms we saw in class should be really useful to extract the pattern from this dataset. Also, since there are a lot of features, it could be a nice playground for classic dimensionality reduction techiniques and also neural embeddings, as well as much of the regularization techiniques we have available for the models we use. Ensemble techiniques are usually successfull in this kind of problems, so it could be a nice opportunity to work with stacking models.

Benchmark Model

The benchmark model is one similar to the one used in most insurances companies in Brasil, which comes down to simple linear models with variable selection and categorization of features. This could be replicated for this dataset as a banchmark, and its performance could be measured in the same way as the proposed models.

Evaluation Metrics

The performance of the solution will be assessed using the Gini Coefficient[5], which can be computed as:

$$Gini = 2 * AUC - 1$$

Where AUC is the area under the ROC curve as a proportion of the total area (which is one). In practive this means that the Gini Coefficient is the proportion of the area under the ROC curve and the total area, but only considering the region above the random line in the ROC. The Gini Coefficient of 1 indicates a perfect model, and of zero indicates one that performs as well as random guess.

Project Design

The project will follow in the standard data science project way. It will start by taking a look at the properties of the features and the target, which includes their distributions and their correlations/joint distributions. The number of missing values for each feature will also be inspected, which will lead to decisions regarding data imputation and discarding of examples or features.

After the initial exploratory analysis, the benchmark model will be reproduced. A logisitic regression with a previous feature selection will be testes, with the minimum amount of feature engineering as possible. This will be the minimum that a model will have to perform.

After this, several simple models will be trained: boosting, bagging, k-nearest-neighbors, feed-forward neural networks, support vector machines, random forest, and possibly much other will be inspected and compared to the benchmark. During this phase, we will keep the feature engineering to a minimum, so it won't interfere with the performance obtained by a method per-se. While doing this, it is possible that a better undertanding of dataset will be obtained.

In the next phase, several techiniques for feature engineering will be applied and tested agaist some of the models decribed before. This include adding new encodings to categorial variables, different ways of normalizing the continous features (standard or robust scaling for example), methods for missing data imputation, and several other possible transformations to these variables. Alongisde the feature engineering, there will be an effort to reduce the dimensionality of the dataset by using PCA, SVD and embeddings, as this should help simpler models. Techiniques to deal with class imbalance, such as over and under sampling, will also be applied to circunvent the fact that only few examples of claims where seen.

The next phase involves tunning the algorithms used before, to find both the best hyper parameters and to decide whether or not to use regularization. This step will certainly include a cross validation procedure to find the best configurations without overfitting.

After that, an effort to train a more complex neural network, staring with a deeper feed-forward network with an embedding layer to encode some of the categorical features. Then techiniques to regularize it will be applied (dropout, batch normalization and others). Maybe this could help by capturing the patterns on non-linear relationships between the features. Better feature representations could also be obtained in this step.

After exploring all these models, the final phase will consist of ensemble techiniques. These models use the predictions of several other models as input to make a new prediction. This can help with some problems as overfitting, in the sense that it will minimize damages by combining a partially overfitting model (one that overfits, but only in certain patterns) to one that isn't overfitting in the same context.

Finally, a report will be written, better describing the data, the final models and the results obtained by the procedure outlined above. This report will then be submitted to the Udacity platform for assessement.