# Porto Seguro's Safe Driver Prediction*

Solomon Gebreyohannes

December 4, 2017

Udacity Machine Learning Engineer Nanodegree Capstone Proposal

## 1 Domain Background

According to Porto Seguro [1], one of Brazil's largest auto and homeowner insurance companies, nothing ruins the thrill of buying a brand-new car more quickly than seeing new insurance bill. The sting's even more painful in case of a good driver. The company believes that it is not fair that one pays so much if he/she is cautious on the road for years. Inaccuracies in car insurance company's claim predictions raise the cost of insurance for good drivers and reduce the price for bad ones. The goal is to utilize machine learning techniques to predict insurance claims with minimum inaccuracy. I propose this challenge entirely taking from Kaggle competition [2], since it is listed as one of the references to get capstone project ideas at Udacity Machine Learning class material. The data needed to train and test the machine learning algorithms are also provided from the same source, Kaggle [3].

There are several related work on machine learning applications. Scikit-learn [4] provides efficient tools for data mining and analysis. Python scripts and documentation (with examples) are available for classification, regression, clustering, dimensionality reduction, model selection, and preprocessing. Panos and Christof summarize machine learning approaches and tools in [5]. They also present a case study of using machine learning for software code analysis. Tziridis et al. used different machine learning techniques to predict airfare prices [6]. Data collection, feature selection, model selection, and evaluation for the case study are discussed in the paper. Using a Kaggle competition challenge, Sangani et al. demonstrate linear regression and gradient boosting to minimize Zillow property estimation error [7]. Similar to regression, there are classification techniques that predicts a probability distribution over a set of classes instead of outputting where an object or instance belongs to. Garg and Roth discusses probability classifiers in [8]. This topic is also discussed in Wikipedia [9].

*Source: The challenge and data sets are taken from Kaggle competition [2,3].

## 2 Problem Statement

The challenge is to build a model that predicts the probability that a driver will initiate an auto insurance claim in the next year. This is a prediction problem since it asks a probability, which is a continuous variable over [0,1]. However, the data for training is given in a classification structure, i.e., policy holders (with given features) vs. whether or not they filed an insurance claim last year. Therefore, it is a probabilistic classification problem. A probabilistic classifier predicts a probability distribution over a set of classes, rather than only outputting the most likely class that the observation should belong to [9]. The features, which will be inputs to the classifier, are grouped into ind, reg, car, and calc. They are also mixed in type - continuous or ordinal, categorical, and binary. Some values of the features are missed from the observation, which are indicated by -1.

## 3 Datasets and Inputs

The datasets contain training and testing data. They are taken from Kaggle competition, available at [3].

### 3.1 Data Descriptions

The data for training is given in a classification structure, i.e., policy holders (with given features) vs. whether or not they filed an insurance claim last year. The features are grouped into ind, reg, car, and calc. They are also mixed in type  continuous or ordinal, categorical, and binary. Some values of the features are missed from the observation, which are indicated by -1.

### 3.2 File descriptions

Training and testing data are available in csv form. The training data, train.csv, contains policy holders information vs. whether or not they filed an insurance claim last year. The test data, test.csv, contains only policy holders information (without the information of insurance claim).

### 3.3 Feature Scaling and Selection

There are 57 total features, which are 26 continuous features, 14 categorical features, and 17 binary features. The categorical features will be changed to binary features. After transforming the categorical features into binary features, the data will have features with only continuous and binary values. There are a lot of features. Since the numbers of training and testing instances are high too, feature selection and reduction are necessary and will be used.

# 4 Solution Statement

This is a probabilistic classification problem since it asks the probability of the insurance claim, which is a continuous number between 0 and 1, and training data has classification nature. Therefore, classification methods (that output probability distribution) will be used to solve it. The target label is provided along with the features in the training data, i.e., insurance claim is filed or not for a given set of features. Therefore, a supervised model will be trained based on this.

Some probabilistic classification methods will be examined. Linear models, Gaussian Nave Bayes, SVM, ensemble, and neural network models will be tested. I will concentrate on one or two for further refinement - tune the best model. One solution is to use grid search (GridSearch) to find hyper-parameters that gives the best model.

# 5 Benchmark Model

The benchmark model chosen is random forest from sklearn. The performance of my model will be compared against the benchmark model at each refinement level.

# 6 Evaluation Metrics

The result is evaluated using the Gini Coefficient. The Gini coefficient is usually defined mathematically based on the Lorenz curve, which ranges from 0 to 1 [10]. Here, however, the Gini Coefficient ranges from approximately 0 for random guessing, to approximately 0.5 for a perfect score. The scoring algorithm compares the cumulative proportion of positive class observations to a theoretical uniform proportion. The code to calculate Gini Coefficient can be found at [11].

# 7 Project Design

The following steps will be followed:
Step 1 Data pre-processing

a. Manage the missing data, which are indicated with values of -1. Fill with random number within the input space for each column.

b. Scale and select features: use PCA for dimension reduction.

c. Split training data to training (80%) and testing (20%) data

Step 2 Choose model

a. Train model on the 80% of the training data. Supervised learning techniques such as linear models, Gaussian Nave Bayes, SVM, ensemble, and neural network are tested and compared.

Step 3 Refine the model(s)

a. After choosing one or two models that outperform most of the models, tune the model(s) (on the 20% of the training data). One solution is to use grid search (GridSearch) to find hyper-parameters that gives the best model. Ensemble method and neural network model perform better.

b. Test the model on the test data

Step 4 Report the result and document the process

a. Save and report the result in csv format.

b. Document the assumptions taken, model used, parameters tuned, result obtained, and the process as a whole.

# References


[1] Porto Seguro.
URL: https://www.portoseguro.com.br/
[2] Kaggle Competition.
URL: https://www.kaggle.com/c/porto-seguro-safe-driver-prediction
[3] Data source: Kaggle.
URL: https://www.kaggle.com/c/porto-seguro-safe-driver-prediction/data
[4] Scikit-learn.
URL: http://scikit-learn.org/
[5] P. Louridas and C. Ebert, "Machine Learning," in IEEE Software, vol. 33, no. 5, pp. 110-115, Sept.-Oct. 2016.
[6] K. Tziridis, T. Kalampokas, G. A. Papakostas and K. I. Diamantaras, "Airfare prices prediction using machine learning techniques," $25^{th}$ *European Signal Processing Conference (EUSIPCO)*, Kos, 2017, pp. 1036-1039.
[7] D. Sangani, K. Erickson and M. A. Hasan, "Predicting Zillow Estimation Error Using Linear Regression and Gradient Boosting," $14^{th}$ *International Conference on Mobile Ad Hoc and Sensor Systems (MASS), IEEE,* Orlando, FL, USA, 2017, pp. 530-534.
[8] Ashutosh Garg and Dan Roth "Understanding Probabilistic Classifiers", To appear in ECML'01. URL: http://l2r.cs.uiuc.edu/ danr/Papers/ecml01.pdf
[9] Wikipedia: Probabilistic Classification.
URL: https://en.wikipedia.org/wiki/Probabilistic_classification
[10] Wikipedia: Gini coefficient.
URL: https://en.wikipedia.org/wiki/Gini_coefficient
[11] Normalized Gini coefficient calculation (code).
URL: https://www.kaggle.com/c/ClaimPredictionChallenge/discussion/703