Porto Seguro's Safe Driver Prediction

Wenting Zhao (wxz163530)
Yuemeng Li (yxl160531)
Huida Liu (hxl165030)

Problem Description

In this project, we will predict the probability that an auto insurance policyholder files a claim.

In the train and test data, features that belong to similar groupings are tagged as such in the feature names (e.g., ind, reg, car, calc). In addition, 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.

Related Work

Dataset Description

There are 2 datasets for project, which are train dataset and test dataset, and a sample_submission dataset, which is not used for this project.

-train.csv
(contains the training data, where each row corresponds to a policyholder, and the target columns signifies that a claim was filed.)
Number of Attributes: 59
Number of Instances: 595212
Number of subjects files a claim: 21694
Number of subjects does not file a claim: 573518

-test.csv
(contains the test data.)
Number of Attributes: 58
Number of Instances: 892817

-sample_submission.csv
is submission file showing the correct format.

Techniques

Dimensionality Reduction

Cross Validation

Train-Validation Split

Pipeline

Logistic Regression

Pre-processing Techniques

String to index.
Filter and Delete the record if there is any missing some attributes.
Map every attributes to features.

Solution and Methods

1. Data analysis:

Because every record has 57 attributes, we decided to implement Dimensionality Reduction to dataset, we reduced the number of attributes into (5,10,30,50,57) and use the same classification model and parameters, we got this result (we use area under ROC and Area under precision-recall curve as matric):

2. Model:

Because this is binary classification, where there are two categories, we decided to use linear model. spark.mllib supports two linear methods for classification: linear Support Vector Machines (SVMs) and logistic regression. We choose to use logistic regression in this project. We choose regParam, Array(0.003, 0.05, 0.01, 0.1, 0.3, 0.5) and maxIter, Array(5,10) parameters when building paramGrid.

3. We use ParamGridBuilder to build the paramGrid

```
val paramGrid = new ParamGridBuilder()
  .addGrid(lr.regParam, Array(0.003, 0.1, 0.05, 0.01, 0.3, 0.5))
  .addGrid(lr.maxIter, Array(5,10))
  .build()
```

4. Additional work
We used cross validation, and set k to 5

Train-Validation Split

TrainValidationSplit only evaluates each combination of parameters once, as opposed to k times in the case of CrossValidator. It is therefore less expensive.

Results

PCA

number of attributes	10	15	20	30	40	50	57
area under ROC	0.56927	0.57914	0.58027	0.61159	0.62047	0.62457	0.5
area under precision -recall curve	0.04777	0.05034	0.05061	0.05634	0.05879	0.05931	0.51822

From the table, we know that it it better to use almost all attributes to train our model.

Train-Validation Split
PCA(10):
Area under ROC = 0.56734
Area under precision-recall curve = 0.04738

PCA(15):
Area under ROC = 0.57874
Area under precision-recall curve = 0.05018

PCA(20):
Area under ROC = 0.57995
Area under precision-recall curve = 0.05048

PCA(30):
Area under ROC = 0.61079
Area under precision-recall curve = 0.05615

PCA(40):
Area under ROC = 0.62028
Area under precision-recall curve = 0.05891

PCA(50) :
Area under ROC = 0.62312
Area under precision-recall curve =0.05930

PAC(57):
Area under ROC = 0.5
Area under precision-recall curve =0.51822

Logistic regression
Our result shows that all of the 892816 subjects in our testing data do not tend to claim insurance, where 98.89% of subjects have the probability of claiming insurance between 0.03 to 0.04 and others have the probability between 0.04 to 0.05.

Conclusion and Analysis

The data has 57 attributes, so we applied PCA to the data. We found out that less attribute leads to lower ROC, so we kept 50 attributes of the data.

After PCA, we consider keeping 50 parameters in the logistic regression model leads to best prediction. We finally get the score of 0.25051 (normalized Gini Index, 0.3 in total) in the competition.

Contribution of team members

Wenting Zhao (wxz163530) : Test PCA method/build the suitable model and find the compare/find the suitable parameters.
Yuemeng Li (yxl160531) : Analyze sample distribution, fit training data to logistic regression model and predict testing data on the model.
Huida Liu (hxl165030) : Parse the data and make data fit the ML pipeline. Optimize hyperparameters in algorithms and Pipelines.

References

ML Tuning: model selection and hyperparameter tuning:

https://spark.apache.org/docs/2.2.0/ml-tuning.html#model-selection-aka-hyperparameter-tuning

logistic regression:
https://www.medcalc.org/manual/logistic_regression.php

COMPETITION

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

OFFICIAL COMPETITION RULES

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