Machine Learning Engineer Nanodegree

Priyaranjan

December 29, 2016

Proposal

Domain Background

For every business, customer relationship is most critical. Customer satisfaction is a key factor for success of the company. Even though company tries to collect customer satisfaction explicitly by surveys and poll. Most of the users don't respond to these. So, there is a need to come with new ways to implicitly measure customer satisfaction. Due to recent advancements in Big Data and machine learning, companies can store relevant data about the customers and this data can be analyzed using machine learning techniques to get the customer satisfaction.

Resources:

1) Forbes-Machine Learning Is Redefining The Enterprise
2) ZenDesk
3) The Top 10 Banks in Customer Satisfaction for 2016
4) How Machine Learning Can Improve Customer Interaction

Problem Statement

Santander bank, based in Boston with principal market is in the Northeastern United States, is a Banking company of our interest. For banking sector customer satisfaction play a critical role in their revenue. By analyzing customer relationship data, they can identify probability of a customer being unsatisfied and take proactive steps. This problem was posted as a competition in Kaggle. General bank Customer data consists of different features like Number of accounts, type of account, how active customer is with the account, Number of complaints, resolutions if provided any etc. Santander bank data consists of various features of the customer and whether the customer is happy or not with the bank. With the help of this data, I would like to train the model to learn from the previous customer experiences and predict probability of the new customer being unsatisfied with the bank.

Datasets and Inputs

Santander bank provided anonymized customer train and test data as excel files which can be downloaded through Kaggle.

Train Data:

There are 76020 customers, each with 371 features of which one is target variable describing whether customer is satisfied or not. All the features are numerical data, with mixture of continuous and ordinal types. Target variable is a binary variable where one equals unsatisfied customer and zero equals satisfied customer.

Link: https://www.kaggle.com/c/santander-customer-satisfaction/download/train.csv.zip

Test Data:

Test Data contains 75818 customers each with 370 features. All the features are same as the training dataset except for the target variable.

Link: https://www.kaggle.com/c/santander-customer-satisfaction/download/test.csv.zip

P.S: As the target variable is missing, accuracy of the model should be calculated by submitting to competition.

Solution Statement

By observing the data we can say that number of features for each customer is huge. After pre-processing the data, we can look for correlations among the features of the data and reduce the feature set if possible. Even though Target variable in the train data is binary, we want to predict the probability of the customer being unsatisfied. So, the output variable will be a value between 0 and 1 (inclusive). Divide training data into train and validation tests. Then choose regression algorithms and train the model. Pass test data to get predictions and submit the predictions to get the accuracy of the model. Tune the model to achieve higher accuracy.

Benchmark Model

In any business entity, most of the customers will be happy with the service (except for few). Same is the case with the Santander bank, most of the customers of the bank are satisfied. A quick analysis of the train data shows that 96.04 are satisfied. So, will have any accuracy of 96.04 if we predict all the customers are happy. This will be a good benchmark solution for this model. Submission assuming every customer is satisfied gave AOC (benchmark) score of 0.5.

Evaluation Metrics

As number of unsatisfied customers is far less than the satisfied customers it would be inappropriate to use accuracy as a measure. . When is customer unhappy, if probability is more than 0.4 or 0.6? We cannot decide by the predicted probability to decide if customer is unhappy or not. We wish to use different thresholds for different situations. In these types of situations we can use ROC curves.

In Receiver Operating Characteristic (ROC) curves, true positive rate (Sensitivity – when unsatisfied customers are classified as unsatisfied) is plotted in function of false positive rates (1 - Specificity) for different cut off points. Each point on the curve represents true positive rate / false positive rate at that particular threshold. Ideal curve will pass through upper left corner (100% sensitivity). If we calculate the area under the ROC (AUROC), it will give correct measure of who well the model is classifying the satisfied and unsatisfied customers. AUROC ranges from 0 -1, where 0 indicates worst performance and 1 as best performance of the model.

Project Design

DATA PRE-PROCESSING:

- Pandas will be used to read both training and testing data csv files.
- Rows containing missing values will be removed.
- Presence of outliers is checked and depending on the situation outlier will be removed.
- Correlation between the features will be examined. Either dimension reduction (PCA) or feature selection (choosing only important K features) techniques will used to reduce the number of features.
- Each feature data will be normalized and features will one-hot encoded if required.

MODEL SELECTION AND TRAINING:

- Depending on the number of features in the dataset and type of features, Regression algorithms like linear regression, Logistic regression or Ensemble methods like Random Forest, Adaboosting, XGboost will be used to train the model.
- Model will be trained through K-Fold cross validation by splitting data into training and validation data. For each fold, performance of the validation data is calculated using AUROC.

PREDICTIONS AND SUBMISSION:

- Test data will be processed same as the training data and will be passed to the model.
- Prediction values are collected and will be saved as csv file.
- This csv file will be submitted to Kaggle to get the accuracy of the model.