Udacity Capstone Praposal

Proposal

Subject History

Supervised learning is the most explored research area within machine learning, as a result of this are commercial applications such as decision trees for recommendation systems. Users who consume these commercial services are already affected by machine learning techniques. However, not all services exploit their generated data to gain insights and thus improve quality for end users.
One of the factors that most affects customers when consuming a service is the delay. To reduce waiting time, organizations can use recommendation systems, indicating more efficient and faster solutions. This not only helps customers, but also allows companies to focus on other things that require regular human intervention. As an example of the use of this technique, it is possible to cite the company Linx Impulse that provides recommendation systems service to large e-commerce companies in Latin America, through emails, shop windows and personalized search to assist users in decision making. Because of these factors, my personal motivation is to use supervised learning techniques to optimize recurring customer service tasks and make better use of time.
The data source will be: https://www.kaggle.com/c/allstate-claims-severity/data

problem description

Allstate Corporation is the second largest personal lines insurer in the United States and the largest publicly traded. Due to their large size, they have to deal with a large number of complaints that take time when made by a human being.
Allstate is currently developing automated methods to predict the cost and thus the severity of complaints. The problem is to create an algorithm that accurately predicts the severity of the claims. As input, you will receive different variables that agents examine to decide the status of claims. They can be continuous or discrete. Since the target variable is a continuous quantity (the amount to be paid to the customer), it is essentially a regression function. Thus, analyzing the purpose of the insurer in this challenge, it is possible to note the algorithm will make recommendations to users.
As mentioned in [4], there are techniques that filter recommendations to ensure better accuracy. These techniques explain the accuracy and correlations of *features*., which will be valuable for this challenge.

Datasets and Entries

The dataset contains 2 .csv files with information needed to make a forecast. They are:

1. Variables in train.csv and test.csv:

- **id** : the id of a couple of training set questions
- **cat1** to **cat116** : category variables (value range not provided nor column names).
- **cont1** to **cont14** : continuous variables (range of values not provided, nor column names).
- **loss** : The amount the company has to pay for a particular claim. This is the target variable. - In test.csv, the loss is not present as we will predict it.

2. In train.csv:

- Number of lines = 188318
- Number of columns = 132
- Highly relevant, as it is the data we will train on.

3. In test.csv:

- Number of Rows = 125546
- Number of Columns = 131
- Highly relevant as it is the data we tested.

Solution Description

It will be necessary to understand the relationship between *features* 130 (116 + 14) with variable *loss* . For this, the solution will be developed using the *Hybrid filtering* technique , [4] to improve the system results. More specifically, the system should use the method of Feature _-combination_ to determine the similarity of *features* . There are *features* that, due to the curse of dimensionality, can result in *overfitting* . To facilitate the work can make a reduction *features* using PCA. It will also be critical to find correlations between *features*to select the best results. In this data exploration step categorical values of alphabets will be converted into numbers that may be easier to process. We will then test machine learning models to see which one performs best using the Kfold division and finally get the mean square error.
The models tested will be: linear regression, XGBoost and Random Forest (Bagging). To adjust the parameters in XGBoost, we will use Grid Search.

Benchmark

The data source is from a Kaggle competition, so the reference model will be chosen as the competition's best score for the test suite, which appears at 1109.70772 absolute mean error (smaller is better).
The linear regression models, XGBoost and Random Forest (Bagging) were tested and run on the test suite provided in this Kaggle competition. The submission file will be sent to

the kaggle website to verify the score. Then we can also compare our model with the benchmark model hosted by Kaggle.
One object for this job is to be ranked among the top 30% results, ie less than 1442,620036 table error.

Evaluation Metrics

The model prediction for this problem can be evaluated in several ways. Since Kaggle's official evaluation of this project is using absolute mean error, it will be used for model evaluation.

Project Design

To develop the project you will first need to explore the data provided by the competition. In the sequence will be done the cleaning of data, conversion of categorical *features* of alphabets into numbers. Next, a *feature engineer* will be made for a cross-validation set to find better correlations between *features* . Once the data has been modeled the models will be tested. First model to be created is the linear regression, so we will have a linear algorithm. The second model will be that of XGBoost. Finally, the third model will be that of Random Forest, as this will have a bit of bagging. After comparing these 3 models will be analyzed which has the lowest average absolute error, then will be submitted for competition in Kaggle. Regarding the tunnig and evolution of the final model, two metrics will be used, the accuracy and precision of the model.

Bibliography

[1] Kaggle, "Allstate Claims Severity" (2016). https://www.kaggle.com/c/allstate-claims-severity , accessed 12/18/2018.

[2] Machinelearningmastery, "Bagging and Random Forest Ensemble Algorithms for Machine Learning" (2016). https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/ , accessed 12/12/2018.

[3] aws, "XGBoost Algorithm" (2018). https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost.html , accessed 10/12/2018.

[4] Egyptian Informatics Journal, "Recommendation systems: Principles, methods and evaluation" (2015). https://ac.els-cdn.com/S1110866515000341/1-s2.0-S1110866515000341-main.pdf?_tid=0168c993-8aaa-4d01-a5c3-2cc49b72e2c1&acdnat=1545258155_05c496d643f6247ca0125c2fa2ef55ef , access 19/12/2018