Machine Learning Engineer Nanodegree

Capstone Project Report

**I. Definition**

**Proposal**

<https://github.com/zhoux0130/capstone-project/blob/main/proposal.pdf>

**The Background**

Since I want to know how to apply ML in insurance, I searched the Competitions in the Kaggle with keyword “insurance”, I find this topic \_“Allstate Claims Severity”.(<https://www.kaggle.com/c/allstate-claims-severity>)

**The Problem Domain and Statement**

My motivation is to use ML to reduce redundant tasks in the insurance field, maybe the customer can be provided with services more quickly. The Allstate can also benefit from this point.

This project wants to create an ML model to predict claims severity. They gave us the train data with more than 130 features to predict severity. Since we want to predict a continuous target variable (loss) with many categorical features, I define this problem as supervised learning and regression problem.

**Datasets and Inputs**

This project contains 2 csv files. They are:

1. train.csv and test.csv features:

* id: the index of a training set data
* cat1 to cat116: category variables(The company will not publish the customer’s privacy information, so all column names are not provided.)
* Cont1 to cont14: continuous variables
* Loss(The target variable): the amount which the company pay for the customer.

1. In train.csv:
   1. 188318 rows
   2. 132 columns
2. In test.csv:
   1. 125546 rows
   2. 131 columns(the test.csv don’t have the loss column)

https://www.kaggle.com/c/learnplatform-covid19-impact-on-digital-learning/data

**Solution Statement**

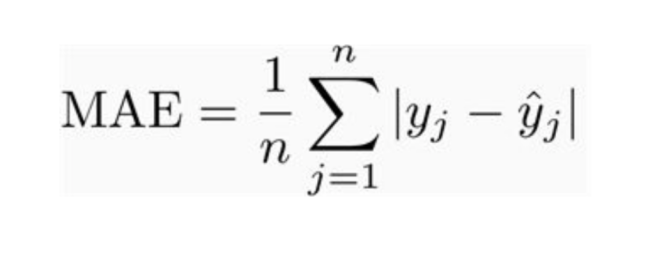
* Exploratory Data Analysis
  + We may use some plot method to see the features ,the correlation between different features.
* Preprocess Data
  + Since we have some categorical features, we need to convert them into numbers. So the model can use them.
  + There are many features in the train set, it may result in overfitting. So we may have to reduce the features by using PCA.
* Choose and Train Model
  + First we can use linear regression as the base model
  + We may also use XGBoost
  + To use Grid Search method test hyper parameters

**Evaluation metrics**

The prediction can be evaluated in several ways. Since the problem is regression type,we can use mean absolute error(MAE), mean squared error(MSE), mean squared log error, etc.

As the competition official evaluation is done by Kaggle using mean absolute error, we will use MAE as evaluation.

**Mean Absolute Error**

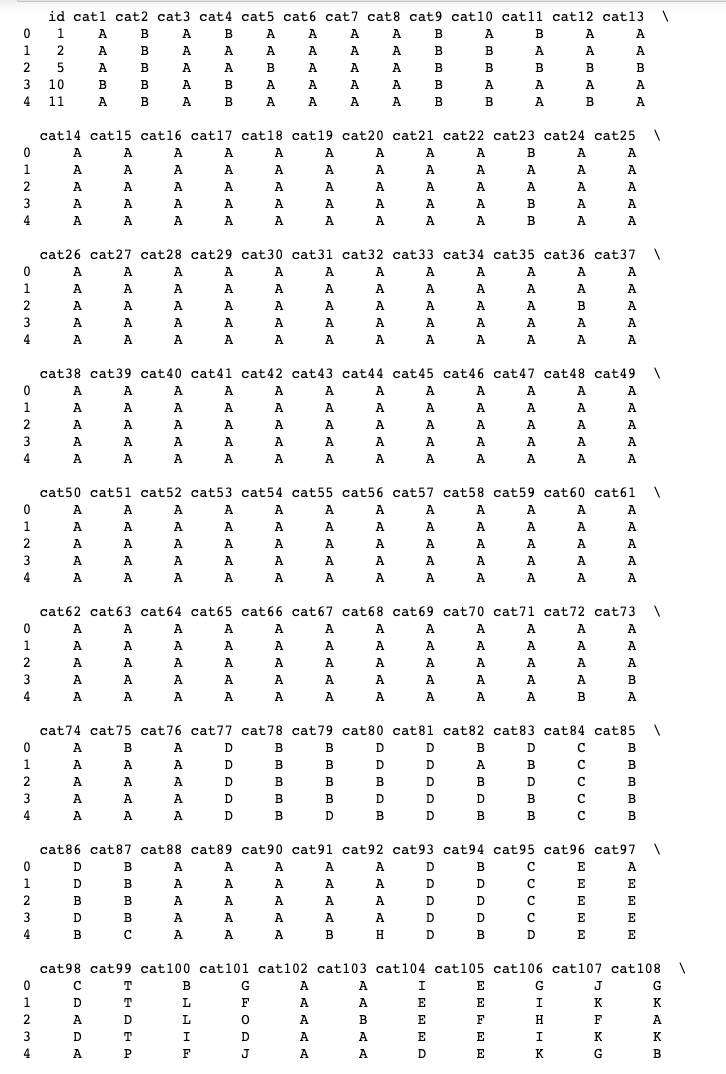
****

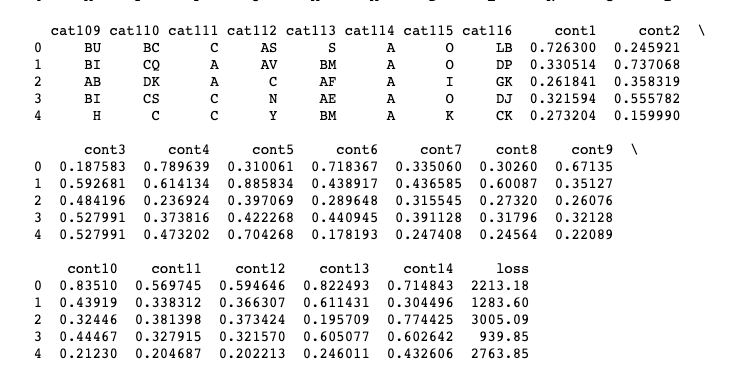
**II. Analysis**

**Data Exploration and Exploratory Visualization**

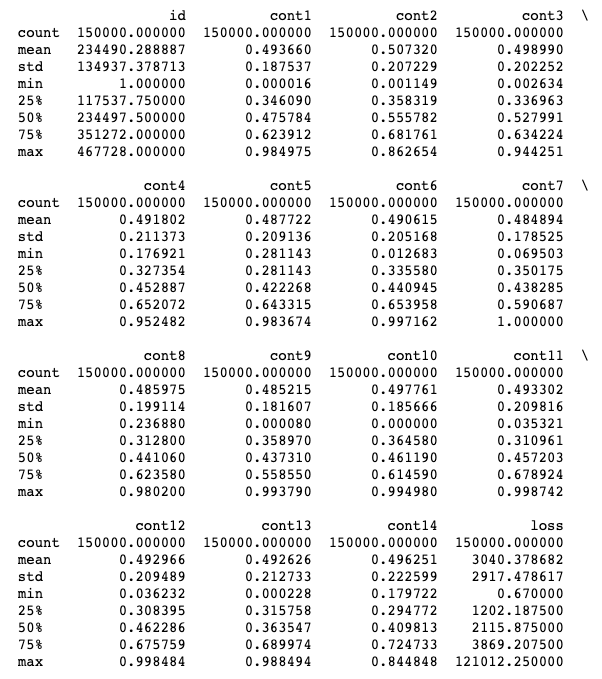
This content is done in file EDA\_1.ipynb.

First we will see how the dataset looks like. They are the first 5 rows in the dataset.

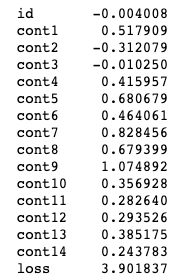




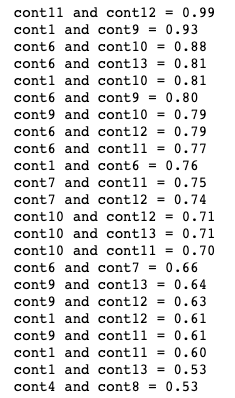
We can see the continuous data’s values are already between 0 and 1, so we can use them directly.



We check the skewness of continuous values. We can see the loss value is positively skewed and needs to be rectified.



Next, we see the correlation between different features, we know the top 5 features have close correlation, we could apply dimensionality reduction method like PCA.



We can also see the value distribution with the categorical features.(The visualization plot can check the EDA\_1.ipynb file)

Cat1 to 72 have only two labels like A and B.

Cat73 to 108 have more than tow labels, but they don’t have too many.

Cat109 to 116 have many labels.

**Algorithms and Techniques**

Since we want to predict a continuous target variable (loss) with many categorical features, I can use the linear regression to solve this problem.

**Linear Regression**

We can fit a curved line passing through feature points, and use it to predict the loss value. But in this problem, we have almost 119 features, it’s too many for the linear regression. We will use this algorithms as base classifier and get a base loss value, and we will try to minimize the loss value with other models.

**XGBoost**

XGBoost is an implementation of gradient boosted decision trees which is for speed and performance. The decision tree take several weak features and combine them to get a strong model. Since I have this project mostly running on my laptop, I like the XGBoost execution speed and model performance.

**Benchmark Model**

We can see leaderboard in the Kaggle project page, the Top solution’s score is 1109.70772 MAE.

Then we can use part of training data as testing data to see the linear regression model’s accuracy as base. Then we compare next model to see how it works. We will choose a better model to run the test.csv, and upload the submission file to check the score.

**III. Methodology**

**Data Preprocessing**

As we just say , loss column is positively skewed. We will take log of the loss + shift.

Shift is another hyper-parameter. It gives better performance. I found it from the discussion(https://www.kaggle.com/c/allstate-claims-severity/discussion/24611)

After log the loss value, it is normalized.

Since there are 114 columns which are categorical features, we will make them numbers.

Also I use 150000 entries as train data and the other 38318 entries as test data from train.csv.

**Implementation**

There are my steps with linear regression algorithms:

1. We import linear regressor from sklearn and also mean absolute error.
2. We train the model
3. We predict the results of the test set  
    Since we transform target value into log(loss + shift), we have to convert the predict result back.
4. We calculate the MAE between the predict values and actual values.

Here are my steps to implement the XGBoost model.

1. We preprocess the data and divide it into training and testing data.
2. We implement the XGBoost model.

I have to notice the tuning processing. Since I have used out my AWS account’s budget, I can’t tune the hyper-parameters on AWS, I have tuned it manually.

**Refinement**

We got a base result(1267.48) with linear regression algorithms, and obtained result(1149.75) with naive XGBoost algorithms, which is better than the base result.

Next we need to find the optimal parameters for the model.Since I have used out my AWS account’s budget, I can’t tune the hyper-parameters on AWS, I have tuned it manually.

The first two parameters which I use are max\_depth and min\_child\_weight. And I run the program on my friend’s cloud computer for once. Finally, I got the better result with the parameters as follow:

Max\_depth = 6 and min\_child\_weight = 8

At last, we use these parameters and reduce the learning rate to 0.01.

**IV. Results**

**Model Evaluation and Validation**

At last, we got the result 1132.3 by tuning the parameters. It’s better than linear and naive XGBoost algorithms.

Then I check the kaggle score, and submit the file. Finally we got the result 1120.234.

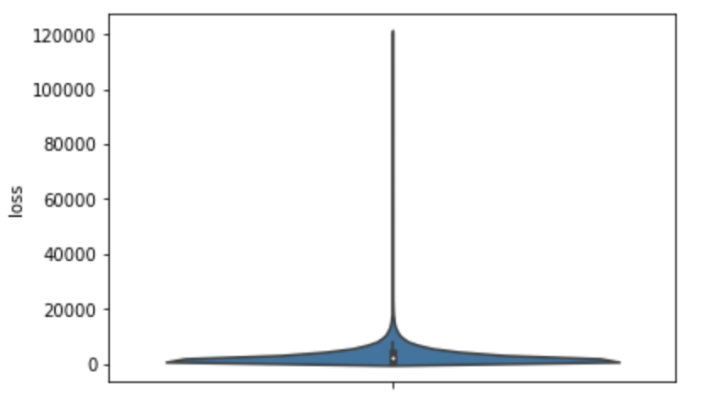
**Justification**

First the score is better than our base algorithms performance. And the result almost is Rank 500 of 3000(total teams), about first top 20% level. I am satisfied about the result.

**V. Conclusion**

**Free-Form Visualization**

We can see the loss value’s violin plot , it can be clearly seen that it is skewed.

****

**Reflection**

Since all the features have no meanings and I don’t have enough domain knowledge, however for sake of academic purposes, I am satisfied with the result.

**Improvement**

1. I could use the AWS SageMaker to train the model automatically**.**
2. PCA could be tired to reduce the features. But we don’t know the features’ meaning, it may bring uncertainty.

**References**

1. Hands-on Machine Learning with scikit-Learn, Keras & TensorFlow, chapter 2
2. Comparing different metric

<https://medium.com/usf-msds/choosing-the-right-metric-for-evaluating-machine-learning-models-part-2-86d5649a5428>

1. Tuning XGBoost

https://www.kaggle.com/c/allstate-claims-severity/discussion/24611