Report of Data Mining(447)
Student: Ning Song
LIN: 802252886

1. Data preparation

- preparing/cleaning your data:

When I get the dataset, I analyze the data and find 8 features in the dataset at first. And through the understand about how to correctly get the point of products, I choose name, lvl1, lvl2, lvl3, descrption, price, type as my initial feature dataset.

And then, I start to remove or complete data with missing value. I use read_csv function in pandas package to transfer the csv to dataframe and use data.isnull().any() to find missing value. And I know that lvl3, descrption, type have that. At first, I use international to fill missing value in type because the international is so many in type and more than words 'local' a lot. Second, I use lvl3 to fill missing value in descrption and use lvl2 to fill missing value in lvl3, but finally I don't do that because the effect of prediction is very bad compared with doing nothing to the missing value.

Then I try to find the data with wrong format and some data with logical errors, but it has nothing. And I do the relevance validation. The result is so small between these features, so I end up my phase of cleaning my data.

- Feature engineering

It is the most important phase in my project. At first, I only use lvl3, price, type as my feature to train the model. Then I use lvl1, lvl2, lvl3, price, type as my feature, and the result of prediction is better. I also try to combine the lvl1, lvl2, lvl3 as new feature of products but the effect of prediction is not good.

After I get so many bad results of prediction and I can't get better result as I combine the lvl1, lvl2, lvl3, I try to use the feature name and descrption. At first, I manually extract the top ten words that appear most frequently among the two features of all individuals as my feature one by one and then train the model. And Although the result of prediction gets better, I find that the result is not so good when I make too many words in name and descrption as feature that I choose or make single word in name and descrption as feature. Because of that, I try to other methods based on the condition that I already make some words that I extract as features.

Finally, I decided to extract the top 40 words from each lvl3 feature as a new feature. And in my experiment, the effect when I use top 40 words is better than top 10,20,30 words. After I extract top 40 words from each lvl3 feature, I build the special feature for every single word: if the name includes this word, the value of the feature is 1. If not, the value of the feature is 0. After doing that, I get so many new features to train my model. The effect is better than before a lot.

And I also try to deal with these new features. At first I use regex to write a function that delete some special symbol such as ' # ' . And then I extract top 40 words again. The result is better than before. But the score did not improve a lot and I don't have chance to deliver that result. All the feature engineering is finished and then I could train my model. This process has taught me a lot because it requires me to keep trying.

2. Models

- Successful models/ algorithms

The first successful model is based on random forest. I use regression decision tree in random forest. The best parameter is:

n_estimators=200,max_depth=75,min_samples_split=65,min_samples_leaf=3,max_features=110,oob_score=True, random_state=100.

The best parameter is based on method of grid search.

The second successful model is based on lightgbm model. The best parameter is:

'boosting_type':'gbdt','objective':'binary','num_leaves':100,'learning_rate':0.01,'feature_fraction': 0.5,'bagging_fraction':0.8,'bagging_freq':5,'metric':'binary_logloss','max_depth':7,'max_bin':300

These two models are both successful and I use them to get my prediction. Finally, I think the lightgbm model is better than random forest because of faster training speed and better accuracy so I choose lightgbm model.

- Failed models/algorithms

The failed models have three: xgboost model, bag-of-words model, linear regression.

- Possible causes of success/failure

At first, I want to talk about my causes of failure. The first failed model is linear regression. I just want to use this model to find whether I could get the result. The effect of this model is very bad. I think it doesn't have linear relation so I give this model up. And then I try to use xg-boost model. Someone says this model is very good to do the problem of predicting result of regression. But when I try this model, it is so slow and the effect is not good as random forest. I think the reason why I failed is over-fitting and I didn't use the xg-boost model correctly. So, I give this model up.

And most important thing in this part, I think it is the bag-of-words model. The Bag-of-words model is an expression model that is simplified under natural language processing and information retrieval (IR). Under this model, words such as sentences or documents can be expressed in a bag with these words, regardless of the grammar and the order of the words. I use the feature_extraction module in scikit-learn to create bag-of-words feature. Finally, I could use bag-of-words feature to train the model by lightgbm. I split the training data to train_data and test_data, and then observe the result of test_data prediction. The result is very good. But when I try to deal with the testing data, I found the result is very bad because the model that I trained doesn't include the feature of word in testing data. It only has the feature of training data. So, the result is very bad. I try to improve or solve this problem, but finally, I failed. I give this model up and use my own methods to extract the feature and do the feature engineering.

My successful model has two: the model based on random forest and the model based on lightbgm. These two models is used to predict results. The lightgbm is very good, I think. The decision tree algorithm based on Histogram and has leaf-wise leaf growth strategy with depth limitation. The model based on histogram is better than xg-boost. Level-wise data can split the same layer of leaves at the same time, easy to multi-thread optimization, control model complexity, not easy to overfit. And the random forest is also good to deal with regression problems.

3. Results

- Final results

My final result is 0.45748(private leaderboard), and my final result of public leaderboard is 0.45069. My model has over-fitting problem.

- My interpretations of the results

I think it is the log-loss point. Log-loss losses, also known as Log-likelihood Loss, also known as Logistic Loss or cross-entropy Loss, are defined on probability estimates. This losses can be used to evaluate the probability output of the classifier. The python code is like: score = log_loss() from sklearn.metrics.log_loss.

The logarithmic loss function is calculated as follows:

$$\boldsymbol{L\big(Y, P(Y|X)\big) = -log\ P(Y|X) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\ log(p_{ij})}$$

Where Y is the output variable, X is the input variable, L is the loss function. N is the input sample size, M is the number of possible categories, and y_{ij} is a binary indicator indicating whether the category j is the real category of the input instance x_i. $P_{ij}$the probability that the input instance xi belongs to category j for the model or classifier.

4. Lesson learned

- what you have learned from doing the project

At first, what I learn a lot is feature engineering. It is the most important thing is this project and all project of data mining. And it is also the most interesting thing in all process. Feature engineering refers to the process of transforming raw data into model training data. Its purpose is to obtain better training data characteristics. It includes feature construction, feature extraction and feature selection. I did all these 3 parts in my project. I do feature construction a lot(combine the lvl1, lvl2, lvl3 as new feature of products), do feature extraction(extract the top 40 words from each lvl3 feature as a new feature), do feature selection many times. What I didn’t do is a lot. Actually, I don’t use statistics, and just use my experience. And I don’t use some model to find the relation between some feature. But I try my best to analyze all the feature that project gives and try to extract more information so that I could get better result.

The data cleaning is also important. The data determines the upper limit of machine learning, and the algorithm just approximates this upper limit as much as possible. The data means the data that we get after data cleaning and feature engineering. Data cleaning or data cleansing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. When I was doing this project, I pay my all attention to data preprocessing. And in the preprocessing, what I can do most is data cleaning. Although in the last I only deal with the missing value in type, I learn a lot about data cleaning so that I could do better in next project. Data cleaning involves several phases：data analysis, definition of transformation workflow and mapping rules, verification, transformation and backflow of cleaned data. And data cleaning has a lot thing need to do: remove of complete data with missing value(what I do most), remove or modify the data with wrong format, remove of modify the data with logical errors, remove unnecessary attributes, relevance validations. These method is used in my project.

And then, I learn lots of model that can be used to deal with regression problems. The model that I like is random forest, bag-of-word, lightgbm and xg-boost. By doing this project, I have deep understand about decision tree. It is a kind of supervised learning. The so-called supervised learning is to give a bunch of samples. Each sample has a set of attributes and a category. These categories are determined in advance, so by learning to get a classifier, this classifier can be new to the class. The object gives the correct classification. Such machine learning is called supervised learning. This project give the point of product in training data, so it is called supervised learning. The interesting thing is that I want to use classifier in my project, but the prediction result need me to predict a regression value, which is interesting. So I learned how to use decision tree to do this projects. Then I want to talk about a little model that I think it is very interesting and effective,

LightGBM contains two key points: light is lightweight, GBM gradient hoist. LightGBM is a fast, distributed, high-performance decision tree-based gradient Boosting framework developed by Microsoft, a gradient boosting framework that uses a decision tree based on learning algorithms. It is very useful and fast. As I use it, the time of it dealing with the dataset and getting the model is 1/3 of time of xg-boost and 1/2 random forest. And the result is good, too. In my next projects, I think I will use that again.

Bag-of-word model is very interesting. It has connection with nature language processing. This model can extract a lot of features and is very fast. Bag-of-word model can be understood as a histogram statistics, starting with a simple document representation method for natural language processing and information retrieval. Similar to histogram, Bag-of-word model only counts frequency information and has no sequence information. Unlike the histogram, the histogram generally counts the frequency of a certain interval. Bag-of-word model selects the words dictionary and then counts the number of occurrences of each word in the dictionary.
I could use this model to get a lot of features, which is useful for training model. I will learn more about that after exam because I am very interested in that.

I understand the hint of project. Feature engineering is the most importing part, and I use simple models to do this project. Ensembling is the thing I don’t make full understand, but I use that thing in my project. I will do research about that. In my next project, I will do more such as deal with parameter to solve over-fitting problem.

Finally, I learn a lot by doing this project and have a very valuable experience in doing projects, which is important. I could do more and learn more in data mining field based on this project and this lesson. Thank you so much!