Caterpillar Tube Pricing Prediction with Elastic Net and Tree-based Boosting

Ziye Liu
Dept. of Mechanical Engineering
The University of Alabama
Tuscaloosa, AL, U.S.
zliu66@crimson.ua.edu

***Abstract*—This project is from a Kaggle data competition: 2015 Caterpillar Tube Pricing. The tubes are widely used in in Caterpillar's construction and mining equipment. Those tube assemblies vary significantly in design, dimensions, and materials. Currently, Caterpillar relies on a variety of suppliers to manufacture these tube assemblies, each having their own unique pricing model. The provided information includes tube specs, components, quantities, etc. The objective of this project is to develop a pricing model to predict the quote a supplier will make for a given tube assembly. Two methods were experimented in this project, elastic net and tree-based boosting (xgboost). Special considerations were made to prevent the information leak for prices of same tube assembly but different quantity to conduct a reliable cross validation. The minimum error rule and one standard error rule were compared for model selection.**

Keywords—tubest;elastic net; xgboost; cross validation

I. INTRODUCTION

The project is from Kaggle data competition "2015 Caterpillar Tube Pricing" [1]. The data was provided by Caterpillar. Caterpillar manufactures an enormous variety of construction and mining equipment. Each machine relies on a complex set of tube assemblies to function. The tube assemblies in Caterpillar's diverse catalogue of machinery varies significantly in design, dimensions, and materials. The tubes are manufactured and sold to Caterpillar by its suppliers. Each supplier has their own unique pricing model. The objective of this study is to develop a model to predict the quote price given the detailed information of the tube assemblies including information about the tube, components attached to the tube, and annual volume datasets. Elastic net and xgboost were experimented in this project. A group shuffle strategy was used during cross validation. Two different model selection criterions, i.e., minimum error rule and one standard error rule were compared.

II. PROBLEM STATEMENT

A. Tube assemblies

Fig. 1 shows an example of tube assemblies. Each assembly is composed by one or more components. The major component is the tube. The specifications to describe a tube include the diameter, length, wall-thickness, number of bends, bend radius, and end types. The tube may have several components attached to it. Depending on the component type, the component may have different attributes.

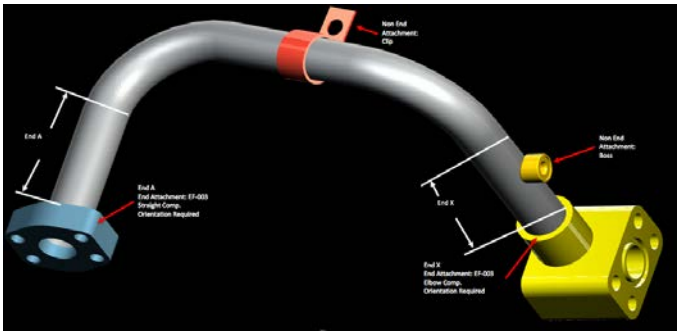

Fig. 1 An example of tube assembly [1].

B. Data description

All the data provided in the competition is shown in Fig. 2. The data is provided in multiple tables. The target/response is the cost which is provided in the "Train" table. Other information provided in the "Train" table includes the supplier, the quote date, the annual usage, minimum order quantity, bracket pricing (a binary variable to show if the quote is given by a bracket format), and quantity. If the cost is given by a bracket format, the tube assembly pricing is given for a multiple levels of purchase quantity. If the cost is not given in bracket format, the pricing is given for a minimum order of amount. The "Test" table provides the same information for test dataset except the cost is not provided. The training dataset has 30123 records, and the test dataset has 30235 records. It should be noted that since the test dataset does not have cost information, 20% of training set was held out as test dataset for model evaluation and selection. The best model was then applied to "Kaggle test" dataset. Kaggle evaluates the predictions using 30% of the test data set to give a public leader board score and 70% for a private leader board score. The private leader board score is used for final ranking. Before the competition ended, the competition participants can only see the public leader board score to prevent the competition participants to overfit the test data. Since the competition already ended in 2015, the public and private leader board scores are almost immediately provided by Kaggle after a prediction is submitted. However, only a few

submissions were tried to test the code, the cross-validation algorithm, and the model selection methodology in this project to avoid overfitting the test dataset to achieve a better Kaggle ranking.

The detailed information of the tube component is provided by the "tube" table, including the material, diameter, length, number of bends, tube end specifications, number of boss components, number of brackets, and other information.

The list of components and the corresponding quantities attached to a tube is given in the "Bill_of_Materials" table. Each assembly may have up to 8 components. The detailed information of each components can be found in the "Comp_[type]" tables (for each type of component, there is a separate table). Different types of components have different number and different types of attributes. Other tables provided by the competition include "Tube_end_form", "Type_component", and "Type_connection".

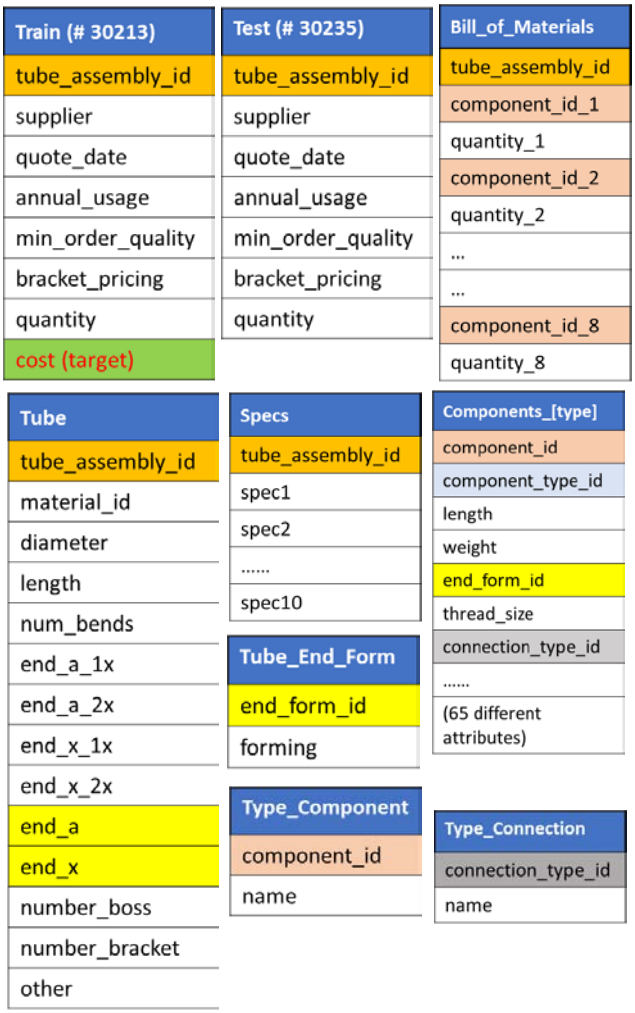

Fig. 2 Data table provides in this competition.

C. Reseach objectives

A reliable quote prediction model is helpful for Caterpillar to select suppliers. The other benefit of a reliable predictive model is that Caterpillar can get a better cost estimation for their product design, which is critical for the design optimization. Thus, this project aims to develop a predictive model for the quote price.

D. Evaluation metric

In this competition, the effectiveness of the predictive model was evaluated by Root Mean Square Log Error (RMSLE), which is given by Eq. 1. Eq. 1 can also be given in the form of Eq. 2. Eq. 2 shows than RMSLE metric is focused on the relative error, i.e., the ratio of the predicted value to the actual value (to avoid numerical issue when the target is too close to zero, 1 is added to the target). Therefore, the same absolute error has more weight for a target of small value, and vice versa.

$$\mathrm{RMSLE}(y_i, \hat{y}_i) = \sqrt{\frac{1}{n}\sum_{1}^{n}[\log(\hat{y}_i + 1) - \log(y_i + 1)]^2} \quad (1)$$

$$\mathrm{RMSLE}(y_i, \hat{y}_i) = \sqrt{\frac{1}{n}\sum_{1}^{n}\left[\log\left(\frac{\hat{y}_i + 1}{y_i + 1}\right)\right]^2} \quad (2)$$

where

n	the number of quotes
$\hat{y}_i$	Predicted price
y_i	actual price
$\log(x)$	the natural logarithm

Eq. 3 shows the conversion of RMSLE to RMSE by applying a logarithm transformation to the target variable. By using this transformation, all the machine learning packages that have RMSE or MSE metric can be directly used in this project.

$$\mathrm{RMSLE}(y_i, \hat{y}_i) = \sqrt{\frac{1}{n}\sum_{1}^{n}(\hat{z}_i - z_i)^2} = \mathrm{RMSE}(z_i, \hat{z}_i) \quad (3)$$

where $z = \log(1 + y)$

III. Data Preprocessing

A. Data preprocessing

Since the data was provided in multiple tables, all the tables were assembled together into a single table for the statistical learning. The most difficult part of the data assembly is to combine "Bill_of_Material" (BOM) table and "Comp_[types]" together because different tube assembly may have different number of components and each component may have different number and different types of attributes. In the combined BOM and components table, each component type has the same number of columns as in the corresponding "Comp-[type]" table. Then indexing the component id in the BOM table, the information of every component for each assembly can be added to the combined table. It should be noted that for each tube assembly, it might have multiple components with the same

type. For simplicity, the numerical features were added together. The categorical feature is encored as 1 if this feature exists for a certain component and 0 if not exist. For example, if a tube assembly has three boss components. The numeric feature such as length, weight, thread size, etc., were added together. The "base_type" feature will be encoded as the count of the features existing on those three boss components. The actual base type information was ignored. Only the existence of the feature was accounted in the combined table. It is certain that information loss in this process will have detrimental effect on the prediction accuracy. However, this method reduced the code and final data complexity significantly and saved time during the data preprocessing stage. Other tables can be merged together by indexing the "tube_assembly_id" easily.

There is a large number of "NA" in the data, which means the tube assembly does not have that feature. All "NA" in the table were filled in as 0. Some feature is given in different units. For example, the thread size in nut components are given by inch (which is given by number) and mm (which has a format like M10). All the SI units were converted into English units for consistency.

The categorical type information of the attached components was lost during the data merge process. One-hot encoding was used for all other categorical features. Log transform was applied to target variable and RMSE was used as evaluation metric for the transformed variable.

B. *Feature engineering*

Several new features were constructed based on the understanding of the tube assembly manufacturability, which is listed as follows:

- Cross-section area of tube
- Total number of components
- Total/mean/min/max weight of components
- Total thread length
- Total thread size
- Total nominal size
- Total number of unique feature
- Total number of orientation feature

IV. REGRRESSION

A. *Group shuffle for cross validation*

Most machine learning packages will shuffle the datasets during cross validation. However, the random shuffle of the datasets causes a problem which makes cross-validation lose its capability to estimate test error. In this dataset, if the cost is given in a bracket pricing format, there is several records of the same tube assembly with different quantities. It is highly possible that the random shuffle splits those records into training and validation set, which will "leak" some information related to the tube assembly pricing from training to validation. For instance, the training data contains the prices of a tube assembly for quantities 10, 50, and 100. In the validation data, the prices of this tube assembly for quantities of 25 and 75 need to be predicted. The original purpose of the pricing prediction model is to estimate the prices based on information of the tube assembly and quantities. However, for this case, the prices for quantities of 25 and 75 can be interpolated from the training data. Considering there is no overlapping of tube assemblies in training and test datasets, the random shuffling causes the cross validation always to underestimate the error.

In this project, a group shuffle strategy was used. The records with same tube assembly was treated as a group. The groups were shuffled during the cross validation to make sure the records with same tube assembly would not be split.

B. *Elastic net*

The linear regression model used in this project was elastic net. Elastic net was a linear regression models with L1 and L2 regulizations [2]. Thus, elastic net is combination of lasso (L1 penalty) and ridge regression (L2 penalty). The ratio of L1 and L2 penalty is controlled by the parameter α. The total penalty is controlled by the parameter λ. In this study, the python package *sklearn.linear_model.ElasticNet* [3] was used. It should be noted that *sklearn.linear_model.ElasticNet* has different notations with reference [2] and other elastic net packages e.g., R/glmnet. The notations used by reference [2] was used in this report to avoid confusion.

Five-fold cross validation was used to tune α [0.1-1.0] and corresponding λ. When $\alpha = 0$, it will be completely ridge regression and when $\alpha = 1.0$, it will be completely lasso regression. For each α, an optimal λ was found using cross validation (Fig. 3). Then this process was repeated for all α, an optimal combination of α and λ was found. The optimal α found in this project was 1.0 (complete lasso) and the corresponding minimum RMLSE was 0.51937 (Fig. 4).

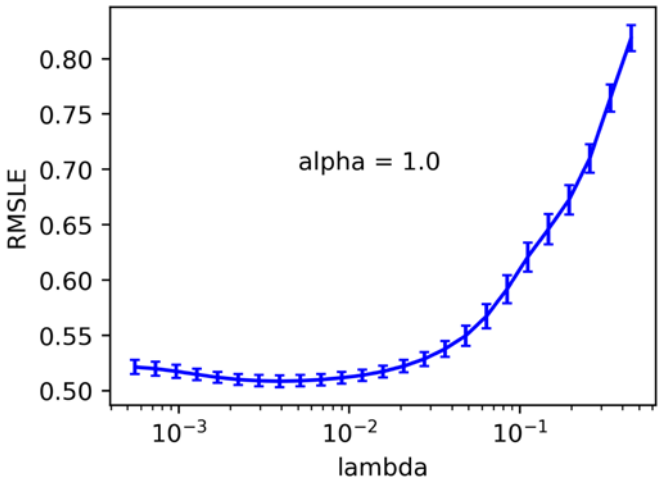

Fig. 3 Finding optimal λ for $\alpha = 1.0$ using cross validation.

C. *Xgboost*

xgboost is an open-source software library which provides the gradient boosting framework for with interface for C++, Java, Python, R, and Julia. It provides a "Scalable, Portable and Distributed Gradient Boosting Library" [4]. It has become one of the most popular algorithms in machine learning community (e.g., kaggle). It is the choice of many winning teams for several recent machine learning competitions. In this study, xgboost python package [5] was used.

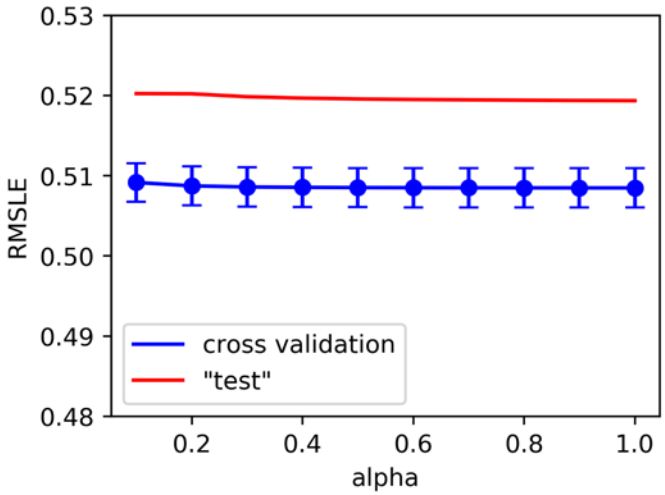

Fig. 4 Optimal α using cross validation and test dataset.

Five-fold cross validation was used to tune max tree depth (4, 6, 8, 10, and 12) and the corresponding number of boosting rounds (the number of trees). The other parameters used in xgboost is shown Table 2. For each max tree depth, the optimal number of boosting rounds (i.e., number of trees) were selected based on minimum cross validation error (min cv) rule and one standard deviation (1se) rule (Fig. 5). Fig. 6 shows the optimal number of trees for different max tree depth selected by the two rules. One standard error rule has more regularization on complexity thus has less number of trees. The selected models were then applied to predict the test dataset prices. Fig. 7 compares the test error of the selected models. Comparing to elastic net, xgboost models have much smaller errors. The models selected by minimum cross validation error rule performs better than one standard error rule. The optimal max tree depth was found to be 6.

Table 2 Parameters used in xgboost

eta	0.05
subsample	0.7
colsample_bytree	0.7
min_child_weight	3
gamma	0.3

The best model was then applied to “Kaggle test” dataset. The public and private leader board RMSLE are 0.23375 and 0.22411, respectively. The ranking for the prediction is 390/1312.

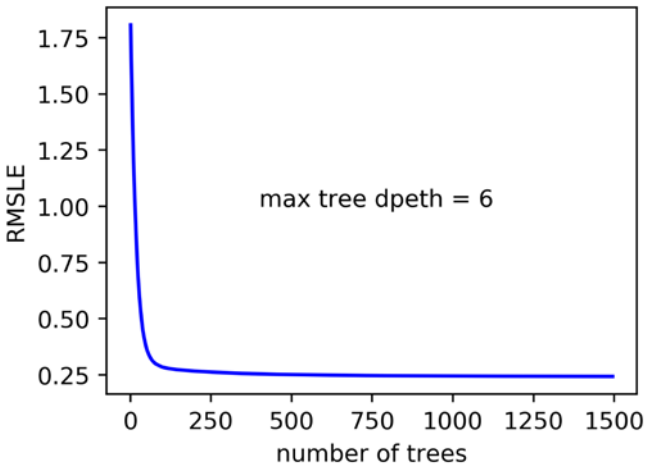

Fig. 5 RMSLE for different number of trees for max tree depth = 6 using cross validation.

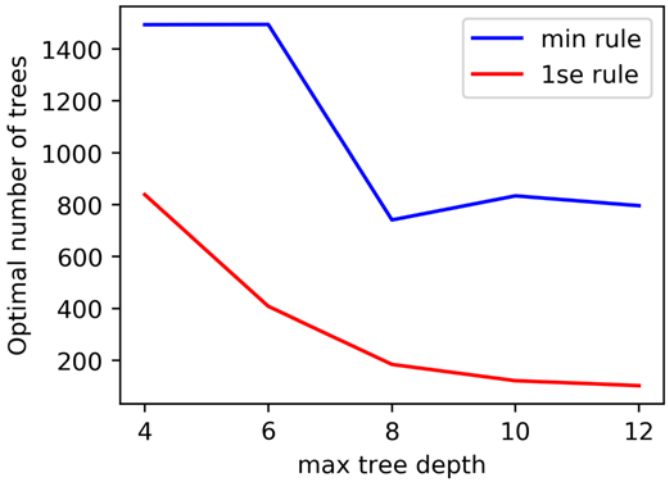

Fig. 6 Optimal number of trees for max tree depth = 4, 6, 8, 10, and 12 selected by min cross validation error rule and one standard error rule using cross validation.

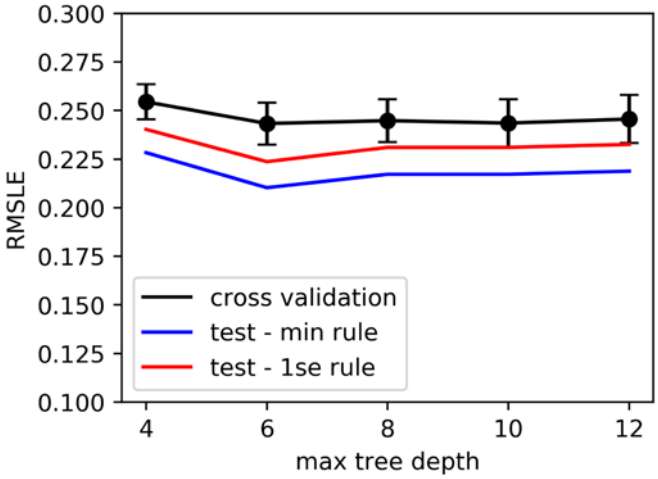

Fig. 7 RMSLE of cross validation and test dataset of the models selected by min cross validation error rule and one standard error rule for max tree depth = 4, 6, 8, 10, and 12.

CONCLUSION

This project developed several models using elastic net and xgboost to predict tube assembly quote prices. A group shuffle strategy was used during cross validation. The model selection criterions minimum cross validation and one standard error rules were compared for xgboost model selection. The conclusions are summarized as follows:

- For this dataset, xgboost models has better predictive accuracy than elastic net models.
- For this dataset, xgboost models selected by min cv error rule is better than one standard error rule.

REFERENCES

[1] Kaggle, Caterpillar Tube Pricing 2015, accessed from: https://www.kaggle.com/c/caterpillar-tube-pricing

[2] Zou H, Hastie T. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2005 Apr 1;67(2):301-20.

[3] Python Sklearn ElasticNet package accessed from http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

[4] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining 2016 Aug 13 (pp. 785-794). ACM.

[5] XGBoost python pakage, Python Package Introduction, acessed from http://xgboost.readthedocs.io/en/latest/python/python_intro.html