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

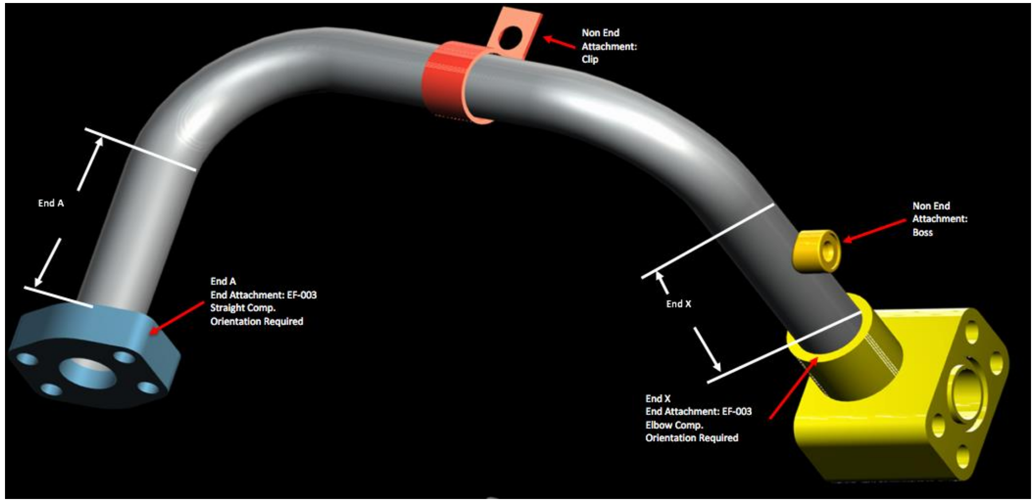
ST697 Project

Dept. of Mechanical Engineering

Ziye Liu

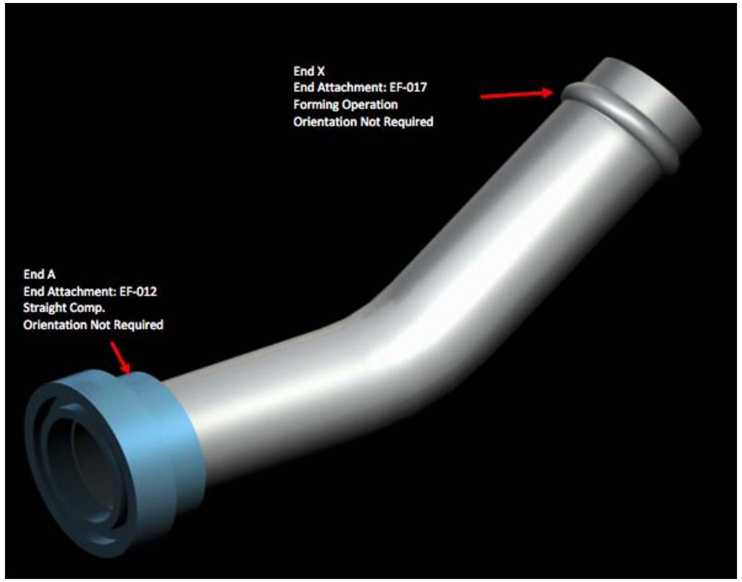
Background

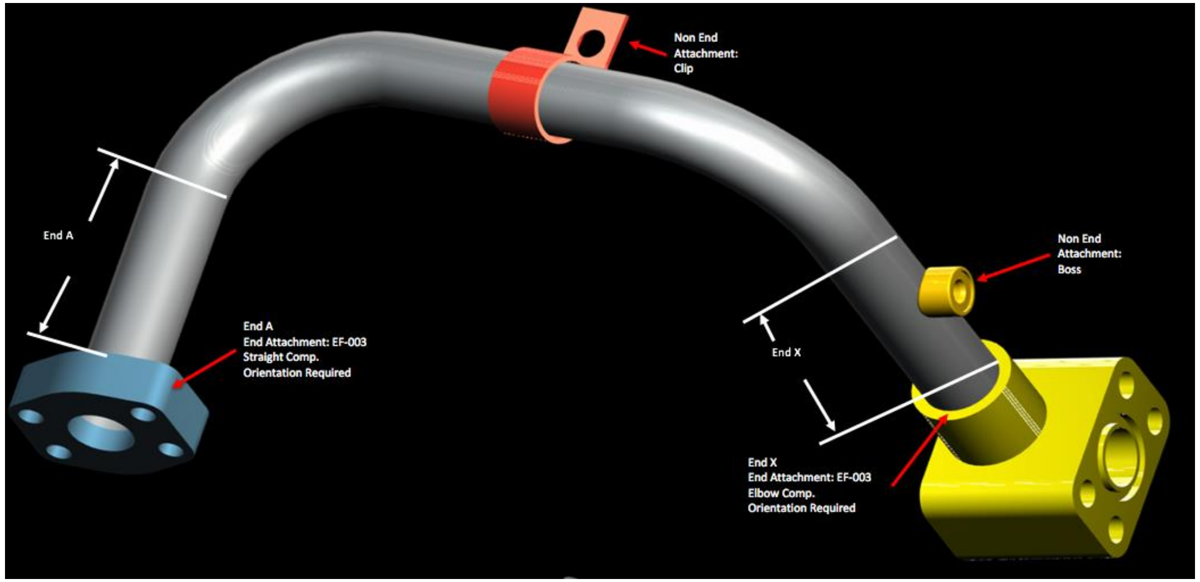
- Project from Kaggle data competition 2015 Caterpillar Tube Pricing
- Caterpillar manufactures construction and mining equipment
- Those equipment use lots of tubes assemblies
- Tubes assembly
  - One or more components
  - Different base materials
  - Number of bends
  - Bend radius
  - End type



Problem Statement

- Given the detailed information of tube, components, and quantity
- Predict the quote price





Data Description

Train (# 30213)
tube_assembly_id
supplier
quote_date
annual_usage
min_order_quality
bracket_pricing
quantity
cost (target)

Test (# 30235)
tube_assembly_id
supplier
quote_date
annual_usage
min_order_quality
bracket_pricing
quantity

Tube
tube_assembly_id
material_id
diameter
length
num_bends
end_a_1x
end_a_2x
end_x_1x
end_x_2x
end_a
end_x
number_boss
number_bracket
other

Specs
tube_assembly_id
spec1
spec2
……
spec10

Tube_End_Form
end_form_id
forming

Bill_of_Materials
tube_assembly_id
component_id_1
quantity_1
component_id_2
quantity_2
…
…
component_id_8
quantity_8

Type_Connection
connection_type_id
name

Components
component_id
component_type_id
name

Type_Component
component_id
name

Components_[type]
component_id
component_type_id
length
weight
end_form_id
thread_size
connection_type_id
……
(65 different attributes)

Evaluation Metric

Root Mean Square Log Error (RMSLE)

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

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

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

More focus on relative error

Convert into RMSE

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

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

Data Preprocessing

	J	K	L	M
_si	thread_pi	nominal_size_1	end_form_id_2	connection
187	12	NA	A-004	NA
312	16	NA	A-004	NA
187	12	NA	A-004	NA
	NA	22.22	A-005	B-002
1	14	NA	A-004	NA
1	14	See Drawing	A-004	NA
	NA	25.4	A-001	B-002

- Assemble all tables together
- Data cleaning
  - Fill all NA values as 0
  - Unified the units (units not consistent, convert SI units into English unit)
  - Fix errors
- One-hot encoding all categorical features
- Log transform target variable and use RMSE evaluation metric

Feature Engineering

- Construct new features
  - Cross-section area of tube
  - Total number of components
  - Total/mean/min/max weight of components
  - Total thread length
  - Total number of unique feature
  - …

Special Consideration for Cross Validation

- If randomly shuffle the datasets during cross validation

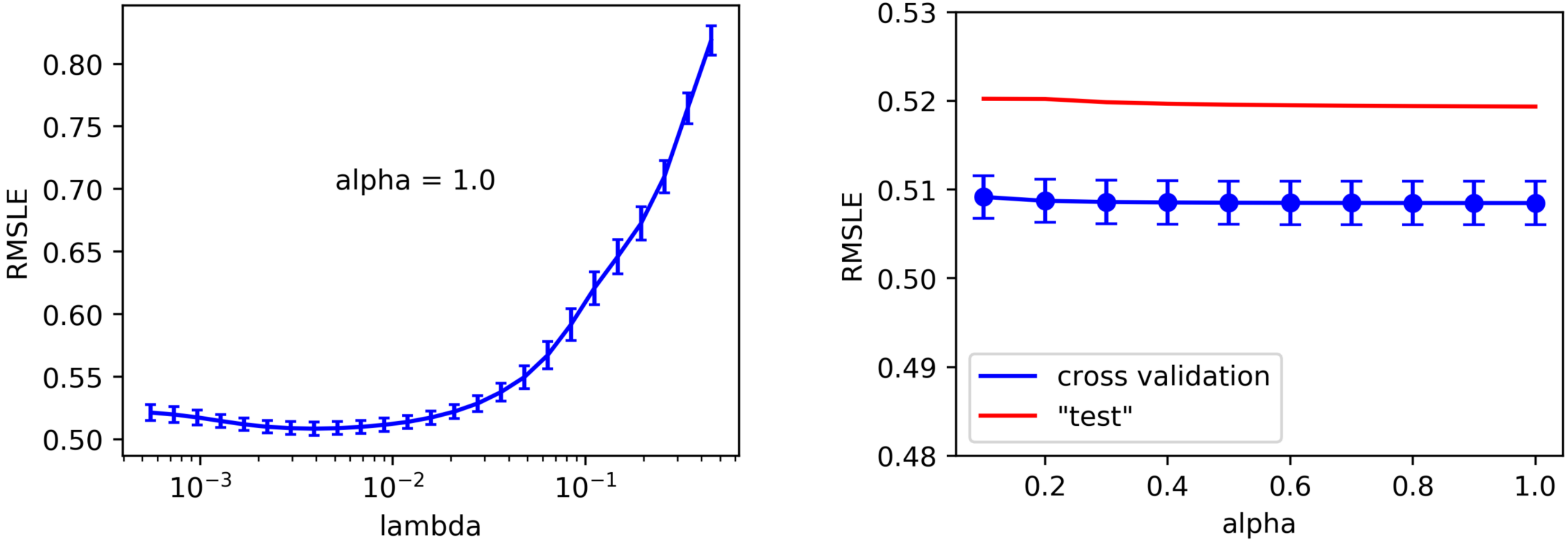
Training			
tube_asembly_id	features	quantity	cost
…			
TA-00056	…	1	28.6468
TA-00056	…	25	5.87570
TA-00056	…	100	5.28034
…			

Validation			
tube_asembly_id	features	quantity	cost
…			
TA-00056		10	
TA-00056		50	
TA-00056		250	
…			

- Cross validation would not work
- Treat records with same “tube_assembly_id” as a group
- Shuffle the groups

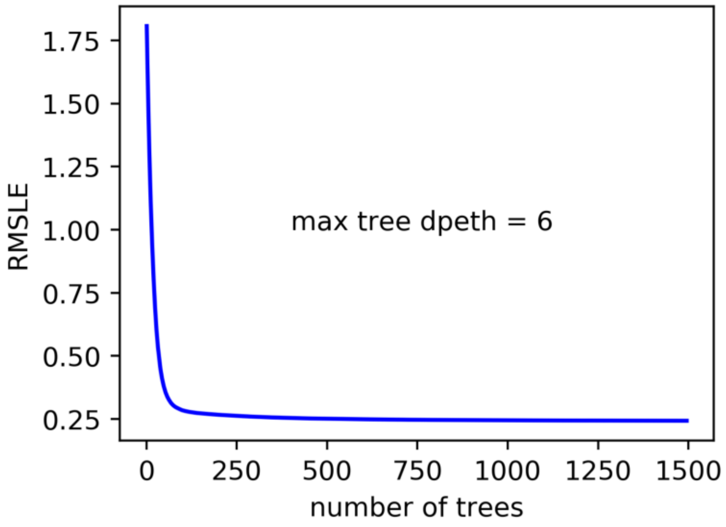
Linear Model (Elastic Net) Prediction

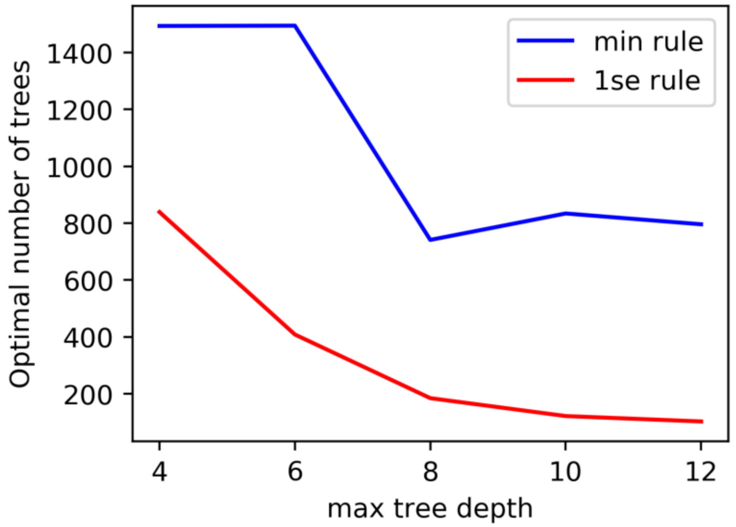
- Grid search + cross validation to find optimal α and λ
- Optimal α = 1.0 (complete lasso) with test RMLSE = 0.51937

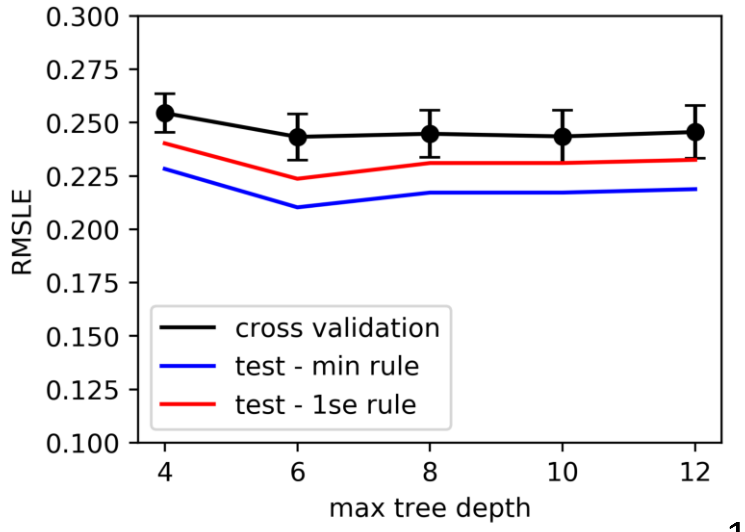


Boosting Model (xgboost) Prediction

- Grid search + cross validation to find optimal max tree depth and number of boosting rounds (i.e., number of trees)
- Min cv error rule performs better than one standard error rule
- Optimal max tree depth = 6, number of boosting rounds = 1495







Kaggle Ranking

- Using best xgboost model selected by min cv error rule
- Leader board results
  - Public leader board RMSLE: 0.233753
  - Private leader board RMSLE: 0.22411
  - Ranking: 390/1323

Conclusion

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

Thanks!
Any questions ?