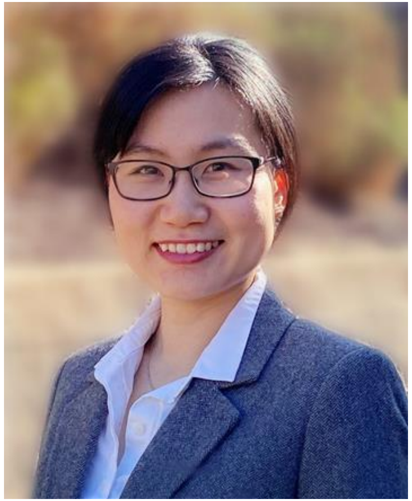
kaggle

Kaggle Actuarial Loss Prediction

Presenter: Yi Li

PRISM Public Risk Innovation, Solutions, and Management

About me



- **Yi Li**
- Sr. Data Scientist @ Public Risk Innovation, Solutions, and Management PRISM
- Featured content creator on Towards Data Science, Level Up Programming etc.
  //medium.com/**@yilistats**

Agenda

01 Actuarial loss prediction overview

02 Exploratory Data Analysis

03 Feature selection & engineering

04 Model training methodology

05 Unsuccessful Attempts

06 Final thoughts

Actuarial loss prediction overview

- The Actuaries Institute of Australia, Institute and Faculty of Actuaries and the Singapore Actuarial Society are delighted to host the Actuarial loss prediction competition 2020/21.

Goal

Predict workers compensation claims using highly realistic synthetic data

Dataset

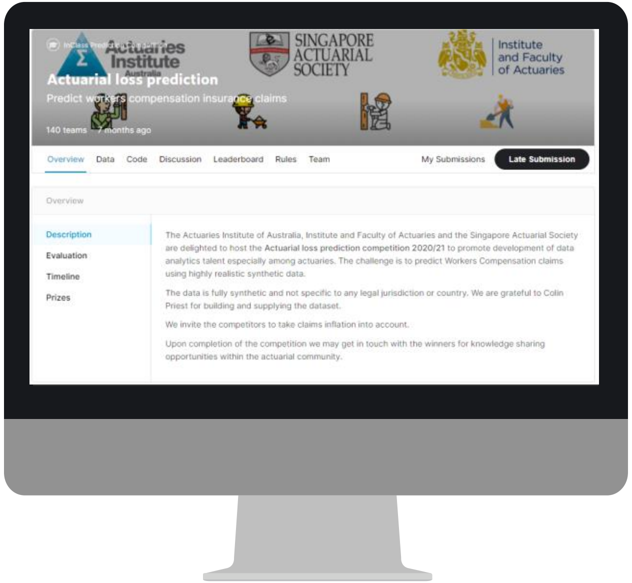
- Training = (54,000, 15)
- Testing = (36,000, 14)

Evaluation Metric

$$RMSE = \sqrt{\frac{1}{N}\sum_{1}^{N}\left(y(i) - \widehat{y(i)}\right)^2}$$

Highlights

- Data is not specific to any legal jurisdiction or country;
- Final winners were determined based on the leaderboard (LB)



kaggle

Actuarial loss prediction overview (*cont'd*)

❖ Sample dataset

	0	1	2
ClaimNumber	WC8285054	WC6982224	WC5481426
DateTimeOfAccident	2002-04-09T07:00:00Z	1999-01-07T11:00:00Z	1996-03-25T00:00:00Z
DateReported	2002-07-05T00:00:00Z	1999-01-20T00:00:00Z	1996-04-14T00:00:00Z
Age	48	43	30
Gender	M	F	M
MaritalStatus	M	M	U
DependentChildren	0	0	0
DependentsOther	0	0	0
WeeklyWages	500	509.34	709.1
PartTimeFullTime	F	F	F
HoursWorkedPerWeek	38	37.5	38
DaysWorkedPerWeek	5	5	5
ClaimDescription	LIFTING TYRE INJURY TO RIGHT ARM AND WRIST INJURY	STEPPED AROUND CRATES AND TRUCK TRAY FRACTURE ...	CUT ON SHARP EDGE CUT LEFT THUMB
InitialIncurredCalimsCost	1500	5500	1700
UltimateIncurredClaimCost	4748.2	6326.29	2293.95

Exploratory Data Analysis

Exploratory Data Analysis

- Duplicated Records

  Current dataset = 0

Missing data handling

MaritalStatus:
assign a new categorical level 'U' ('Unknown')

MaritalStatus	count	median	mean
M	22516	4166.46005	12024.93360
S	26161	2322.79973	9105.57810
U	5323	5828.91805	16009.29959

- Extreme outliers

  UltimateIncurred:
  \$4M → \$1M

```
count      54000.00000
mean       11003.36917
std        33390.99129
min          121.88681
25%          926.33845
50%         3371.24173
75%         8197.24865
max      4027135.93500
Name: UltimateIncurredClaimCost, dtype: float64
```

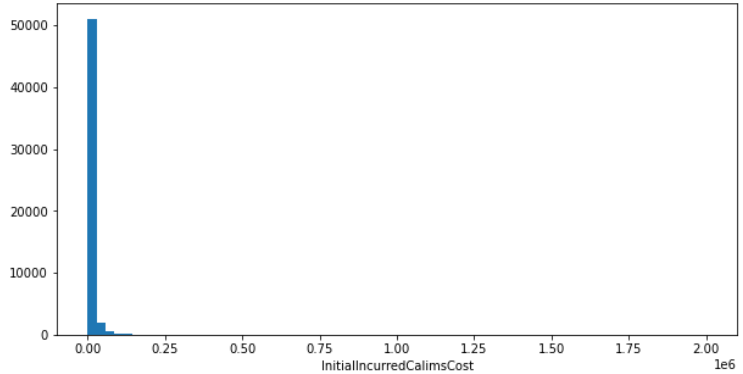
Exploratory Data Analysis (*cont'd*)

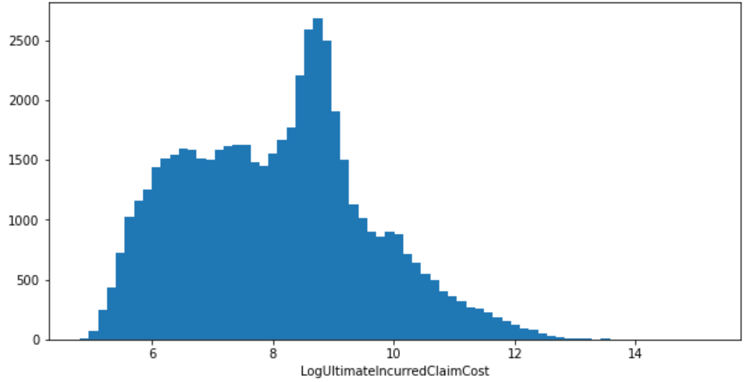
❖ Summary statistics of each numeric variables indicate that standardization is needed.

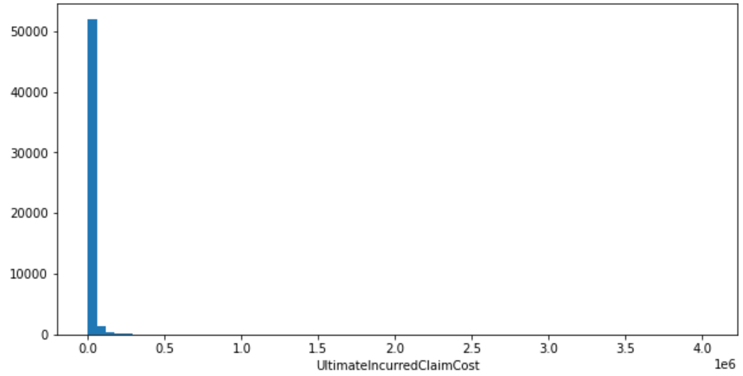
	count	mean	std	min	25%	50%	75%	max
Age	54000.00000	33.84237	12.12216	13.00000	23.00000	32.00000	43.00000	81.00000
DependentChildren	54000.00000	0.11919	0.51778	0.00000	0.00000	0.00000	0.00000	9.00000
DependentsOther	54000.00000	0.00994	0.10935	0.00000	0.00000	0.00000	0.00000	5.00000
WeeklyWages	54000.00000	416.36481	248.63867	1.00000	200.00000	392.20000	500.00000	7497.00000
HoursWorkedPerWeek	54000.00000	37.73508	12.56870	0.00000	38.00000	38.00000	40.00000	640.00000
DaysWorkedPerWeek	54000.00000	4.90576	0.55213	1.00000	5.00000	5.00000	5.00000	7.00000

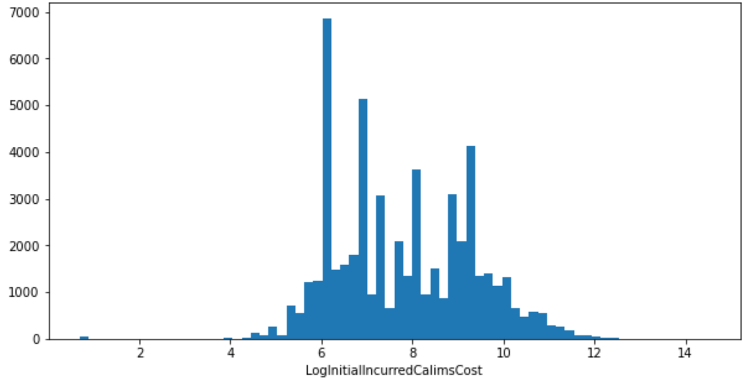
Exploratory Data Analysis (*cont'd*)

- **InitialIncurred** and **UltimateIncurred** are severely right skewed → Log transformation
- Target variable: Log transformed Ultimate Claims Cost









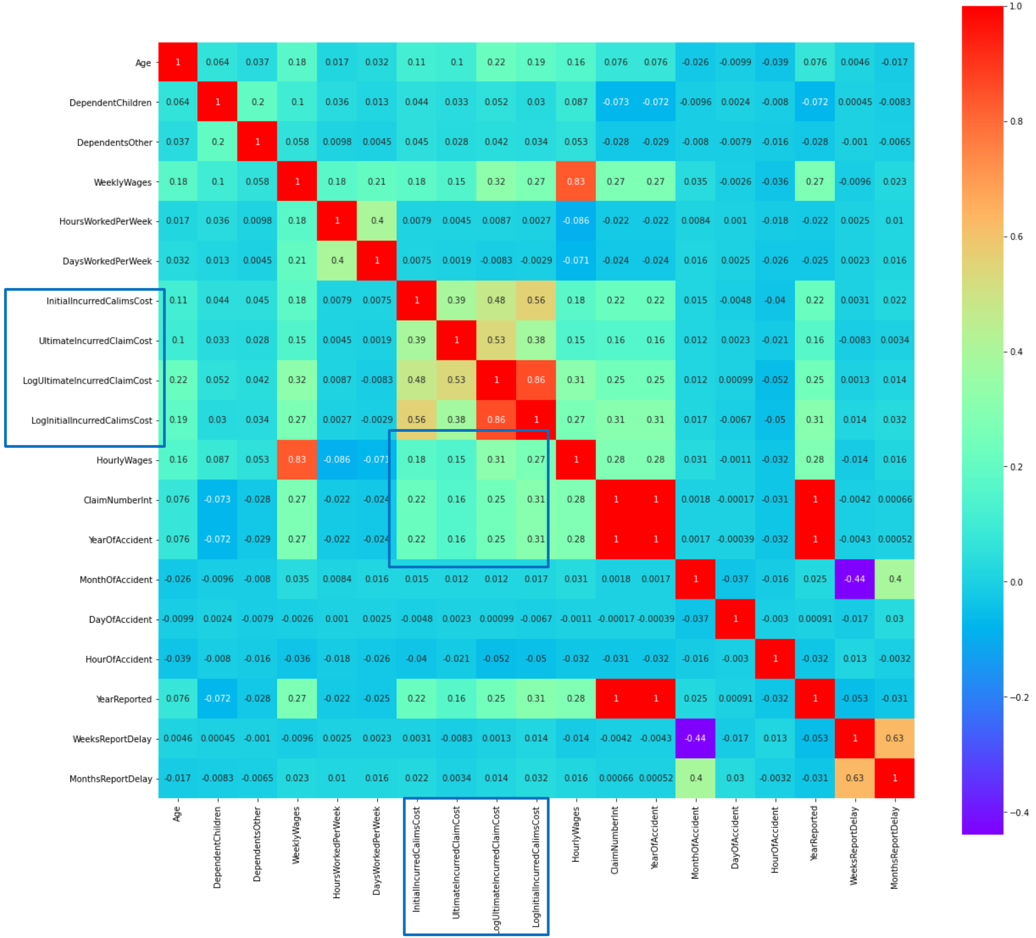
Features Selection/ Engineering

Features Selection/Engineering

- The current model uses both the original features and derived features
- Key derived features:
    - Integer part of the claim number: e.g., WC8285054 → 8285054
    - Year and Month of the Accident derived from the DateTimeofAccident: e.g., 2002-04-09T07:00:00Z → 2002, 04
    - Year the claim reported derived from the DateReported: e.g. 1996-04-14T00:00:00Z → 1996
    - Report delay in Weeks and Months: e.g., week/month of reported – week/month of accident
    - Hourly wages derived from the WeeklyWages

Features Selection/Engineering (*cont'd*)

- Feature correlation,



- Multicollinearity issue?
- No variable has a strong correlation with the (log) UltimateIncurredLoss by itself → interactions

kaggle

Features Selection/Engineering (*cont'd*)

❖ Key derived features: Claims Description

	UltimateIncurredClaimCost	ClaimDescription
11027	4027135.93500	SLIPPED ON WET FLOOR FRACTURED BASE OF HAND
23036	865770.64860	WHILST MASSAGING FELT PAIN SOFT TISSUE INJURY LEFT HAND
37813	823706.30120	TABLE TIPPED OVER SOFT TISSUE INJURY RIGHT HAND
3193	768485.11820	LIFTING BACK BACK STRAIN
923	742003.23350	SHEARING HAND PIECE BLISTER RIGHT HAND
47532	741498.02750	LIFTING PARTS STRAIN BACK LOWER BACK STRAIN
28959	713784.06360	LIFTING BOX FROM TOOL BOX HERNIA
25148	608650.42590	LIFTING DRUM LOWER BACK PAIN

Part of Body: Hand

Cause of Injury: Slip

Part of Body: Hand
Cause of Injury: Unknown

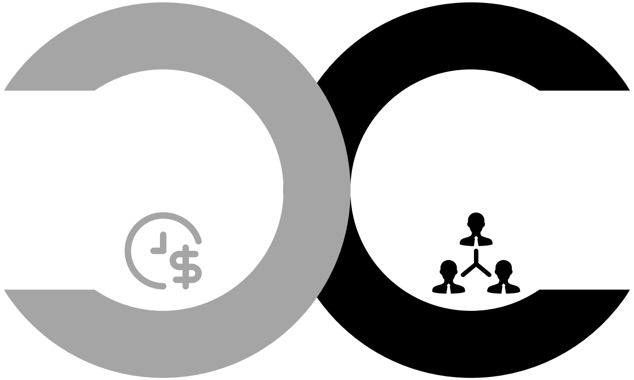
Part of Body: Lower Back
Cause of Injury: Lifting

Features Selection/Engineering (*cont'd*)

❖ How to incorporate the Parts of Body and Cause of Injury into modeling?

Unique Parts of Body = 44

['Unknown', 'forearm', 'external', 'hand', 'foreign body', 'arms', 'hernia', 'eyes', 'conjunctivitis', 'concussion', 'chemicals', 'stress', 'wrists', 'fingers', 'teeth', 'wrist', 'legs', 'disc', 'shoulders', 'hands', 'thoracic', 'abdominal', 'vertebrae', 'fumes', 'knees', 'hips', 'depression', 'achilles', 'biceps', 'cervical', 'trapezius', 'bicep', 'ankles', 'hearing', 'thighs', 'toes', 'lungs', 'feet', 'dizziness', 'anxiety', 'asthma', 'blindness', 'eardrum', 'nausea']

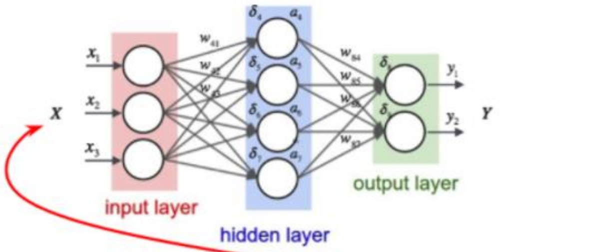


Unique Cause of Injury = 78

['lift', 'step', 'cut', 'dig', 'reach', 'lacerate', 'bruise', 'leave', 'slip', 'strain', 'right', 'pick', 'Others', 'fall', 'use', 'bend', 'hit', 'push', 'walk', 'strike', 'grind', 'catch', 'jam', 'play', 'handle', 'sort', 'weld', 'fracture', 'carry', 'infect', 'drill', 'crush', 'clean', 'twist', 'pull', 'burn', 'move', 'drop', 'blow', 'trip', 'drive', 'deal', 'slice', 'sprain', 'lower', 'turn', 'enter', 'redback', 'strap', 'stand', 'roll',......]

Features Selection/Engineering (*cont'd*)

- One-hot or dummy encoding are sparse for high-dimensional categorical variables,
  - **Cause of Injury** has 78 values. With one-hot encoding, each value will be mapped to a vector containing 78 integers, and 77 are zeros → not computationally efficient;
- Reduce the dimensionality of categorical variables → **Entity Embedding** (Guo & Berkhahn, 2016)



Head	0.4	-0.3	0.6	0.1
Shoulders	0.2	0.2	0.5	-0.3
Knees	0.1	-1.0	1.3	0.9
Depression	-0.6	0.5	1.2	0.7
Forearm	0.9	0.2	-0.1	0.6
Lungs	0.4	1.1	0.3	-1.5
Concussion	0.3	-0.2	0.6	0.0

- Inspired by **word2vec**;
- Refers to the **representation** of categories by n-dimensional numeric vectors;
- Build a Neural Network model to predict (log) UltimateIncurredLoss with each categorical variable;
- Often used as a part of standard training process of Neural Network model, but **weights/vectors** can also be extracted as input features for other machine learning algorithms;

Features Selection/Engineering (*cont'd*)

- Dichotomize InitialIncurred (labeled as Severity):
  - value >= 95% coded as 'High'; values < 95% coded as 'Low'

- Feature transformation prior to modeling

Continuous variables

e.g., Age, WeeklyWages

Standardized with $Z = \frac{x - \mu}{\sigma}$

Categorical variables with fewer levels

e.g., Gender, Marital Status, were one-hot encoded

MaritalStatus_M	MaritalStatus_S	MaritalStatus_U
1	0	0
1	0	0
0	0	1
0	1	0
1	0	0

Model Training Methodology

Model Training Methodology

ORIGINAL FEATURES

- Transformed
- Target: Log UltimateIncurred

DERIVED FEATURES

- Basic feature engineering
- Entity embedding

FIT

MODEL BUILDING & TUNING

- Training: Validation set = 0.8:0.2
- Objective function: MSE
- Hyperparameters tuning

SELECT

PREDICTION

Make final predictions against the testing set

- Training set = 54,000*0.8 = 43,200;
  Validation set = 54,000*0.2 = 10,800;
- 10-Fold Cross-Validation to measure the training performance;
- Bayesian optimization method to select the optimal set of hyperparameters.

$$x^{\star} = \arg\min_{x \in \mathcal{X}} f(x)$$

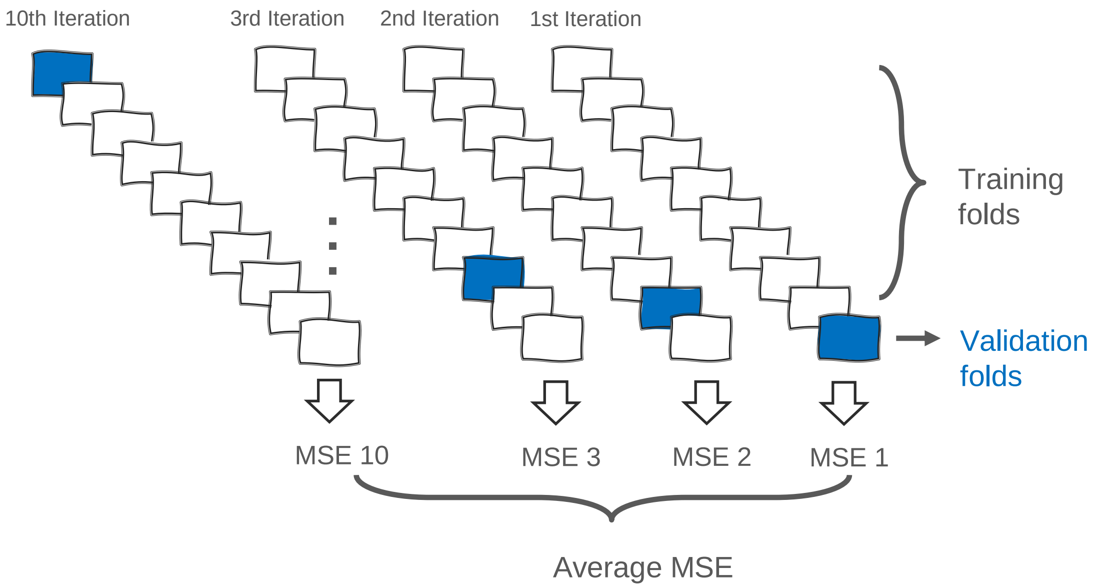
$$P(error|hyperparameters) = \frac{P(hyperparameters|error)P(error)}{P(hyperparameter)}$$

Model Training Methodology (*cont'd*)

The K-Fold Cross-Validation:

- Training data is split into K subsets, e.g., 10 in the current model;
- K models are trained using all subsets but one;
- Performance of each of the K models is tested on the last subset;
- Average to get the final K-Fold performance.

This approach is efficient and universally works regardless of the modeling algorithms.



Model Training Methodology (*cont'd*)

❖ Stacking model with two **Gradient Boosting Machines** (GBM) models – LightGBM and Xgboost – as base learners outperformed.

Algorithm 1 Friedman's Gradient Boost algorithm

Inputs:

- input data $(x, y)_{i=1}^{N}$
- number of iterations M
- choice of the loss-function $\Psi(y, f)$
- choice of the base-learner model $h(x, \theta)$

Algorithm:

1: initialize $\widehat{f_0}$ with a constant
2: **for** $t = 1$ to M **do**
3: compute the negative gradient $g_t(x)$
4: fit a new base-learner function $h(x, \theta_t)$
5: find the best gradient descent step-size ρ_t:

$$\rho_t = \arg\min_{\rho} \sum_{i=1}^{N} \Psi\left[y_i, \widehat{f_{t-1}}(x_i) + \rho h(x_i, \theta_t)\right]$$

6: update the function estimate:

$$\widehat{f_t} \leftarrow \widehat{f_{t-1}} + \rho_t h(x, \theta_t)$$

7: **end for**

Friedman, 2001 Greedy function approximation: A gradient boosting machine

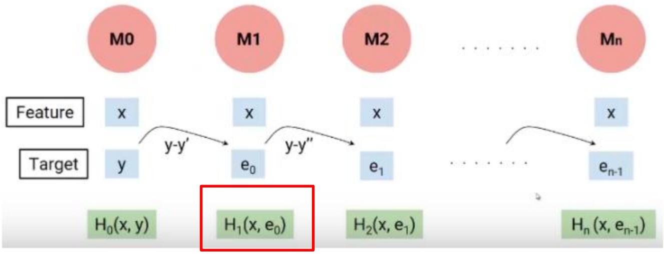


Image Source

Each successive model is built to reduce the errors, a.k.a, *pseudo-residuals*, of all the previous models.

Model Training Methodology (*cont'd*)

❖ Stacking model with two Gradient Boosting Machines (GBM) models – **LightGBM and Xgboost** – as base learners outperformed.

Leaf growth

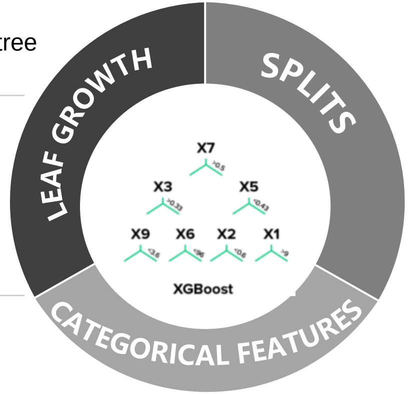
Splits up to the *max_depth* hyperparameter → Prunes the tree backwards

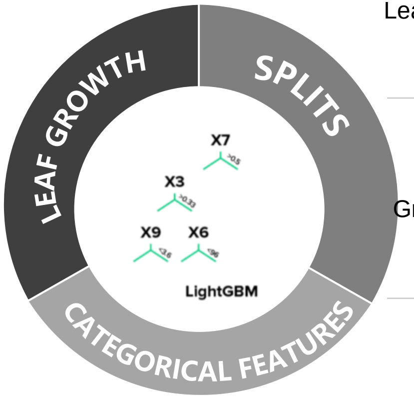
Splits

No weighted sampling techniques are implemented

Categorical features

No built-in methodologies to handle categorical features





Leaf growth

Leaf-wise or best-first tree growth

Splits

Gradient-based one-side sampling (GOSS)

Categorical features

Partition categorical features into 2 subsets according to the training objective

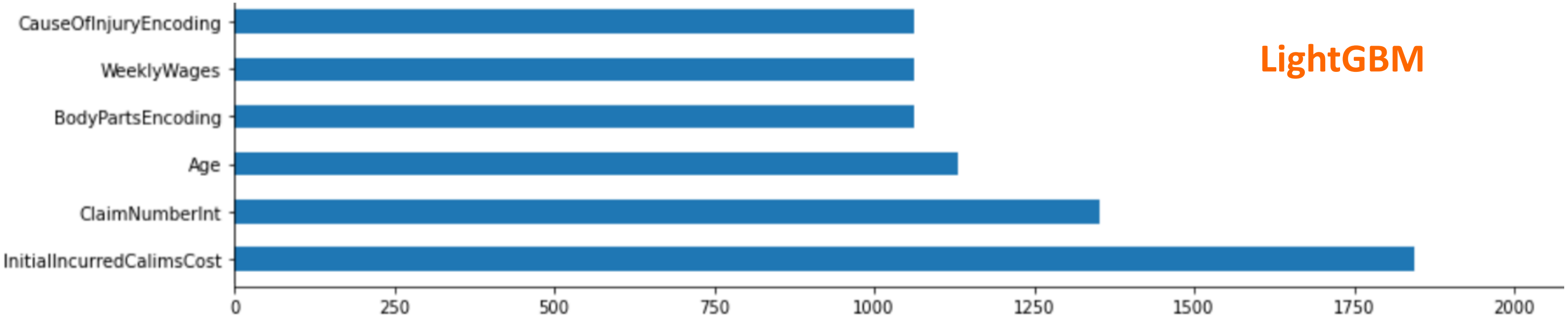
Model Training Methodology (*cont'd*)

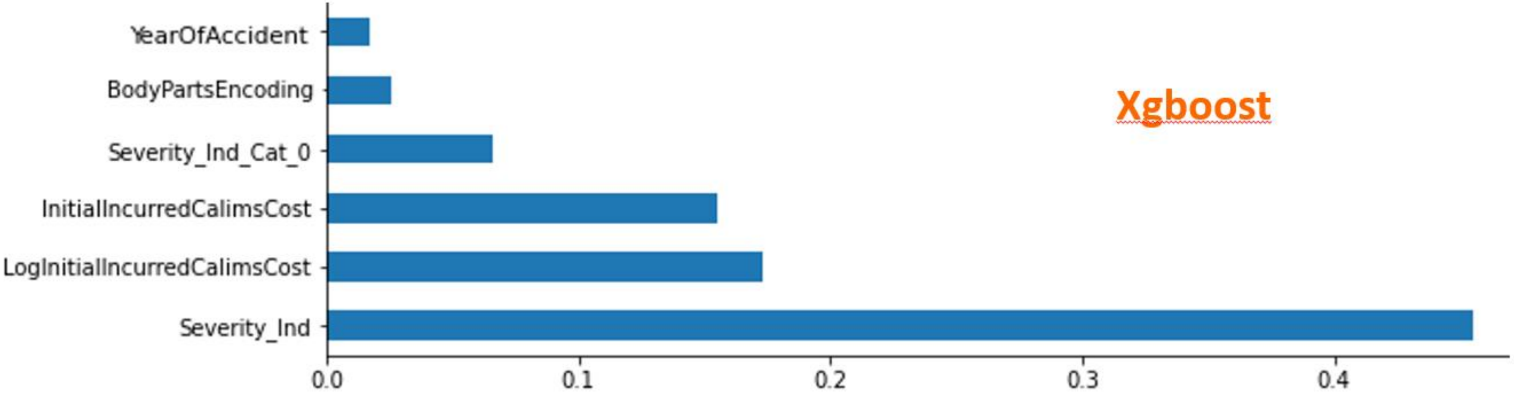
- **Stacking model** with two Gradient Boosting Machines (GBM) models – LightGBM and Xgboost – as base learners outperformed → Greedy search to find the weight to each base learner:

  - Prediction = 0.85 * LightGBM + 0.15 * Xgboost

Model Training Methodology (*cont'd*)

❖ Top 6 features identified by the models,



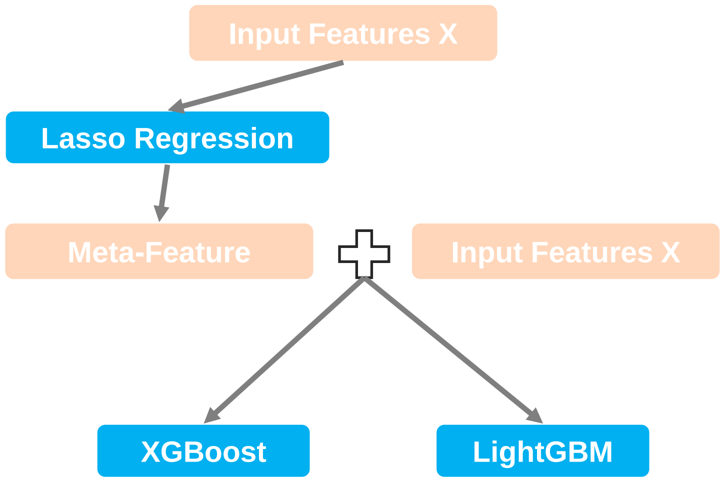


Model Training Methodology (*cont'd*)

❖ Things not worked for the public LB, but worked for the private LB (final standing):

- Ensemble model: Lasso regression to create a meta-feature, and then including this meta-feature as one predictor to the two base learners

- More derived features:
  - Binned Initial Incurred Claims Cost
  - Initial Incurred Cost Per Payroll = Initial Incurred Claims Cost / WeeklyWages



Unsuccessful Attempts

Unsuccessful Attempts

- Latent Dirichlet Allocation (LDA) topic modeling on Claims Descriptions (inspired by the discussions in the Kaggle forum)
    - Also tried to standardize some words (e.g., stemming and lemmatization) in the Claims Descriptions before LDA, e.g., convert ‘strained’ to ‘strain’, convert ‘laceration’ to ‘lacerate’; however, neither seems to work

- Leave One out (LOO) or Frequency encoding for the derived Part of Body variable

- Other derived variables, e.g., Total Dependents = Dependent Children + Dependent Other

- Neural Network model: neither as another base learner nor creating a meta-feature predicted by other features

Final Thoughts

Final Thoughts & Future Explorations

- There's no Evaluation Date in this dataset, hence the Claim Age cannot be calculated. It would be interesting to explore the importance of Claim Age in predicting the ultimate claim loss.

- If the evaluation date is available, it can be used to build out the loss triangle and explore how to incorporate the loss triangle into the machine learning models.

- Other objective functions, e.g., Gamma or Tweedie as they are popular distributions in actuarial loss modeling.

Thank you!

Yi Li

yilistats@gmail.com

medium.com/**@yilistats**

PRISM
Public Risk Innovation,
Solutions, and Management
A Public Agency

Public Risk Innovation,
Solutions, and Management

Appendix

❖ ['WeeksReportDelay', 'MonthsReportDelay', 'MonthOfAccident']