Gradient Boosting

Weijie Zhao

09/20/2022

Why Gradient Boosting?

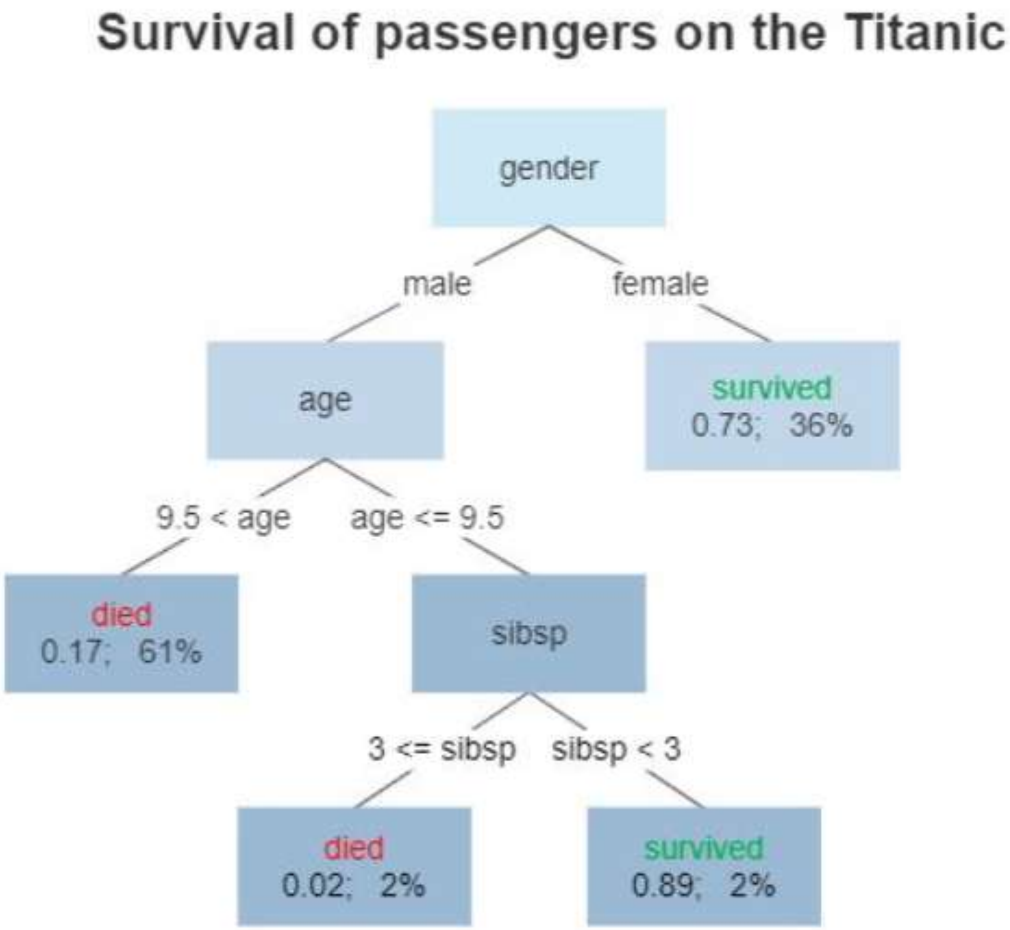
Machine Learning Challenge Winning Solutions

LightGBM is used in many winning solutions, but this table is updated very infrequently.

Place	Competition	Solution	Date
1st	M5 Forecasting - Uncertainty	link	2020.7
3rd	M5 Forecasting - Uncertainty	link	2020.7
3rd	ALASKA2 Image Steganalysis	link	2020.7
1st	M5 Forecasting - Accuracy	link	2020.6
2nd	COVID19 Global Forecasting (Week 5)	link	2020.5
3rd	COVID19 Global Forecasting (Week 5)	link	2020.5
1st	COVID19 Global Forecasting (Week 4)	link	2020.5
2nd	COVID19 Global Forecasting (Week 4)	link	2020.5
2nd	2019 Data Science Bowl	link	2020.1
3rd	RSNA Intracranial Hemorrhage Detection	link	2019.11
1st	IEEE-CIS Fraud Detection	link	2019.10
2nd	IEEE-CIS Fraud Detection	link	2019.10
2nd	Kuzushiji Recognition	link	2019.10
1st	Los Alamos National Laboratory Earthquake Prediction	link	2019.6
3rd	Los Alamos National Laboratory Earthquake Prediction	link	2019.6
1st	Santander Customer Transaction Prediction	link	2019.4
2nd	Santander Customer Transaction Prediction	link	2019.4
3rd	Santander Customer Transaction Prediction	link	2019.4
2nd	PetFinder.my Adoption Prediction	link	2019.4
1st	Google Analytics Customer Revenue Prediction	link	2019.3
1st	VSB Power Line Fault Detection	link	2019.3
5th	Elo Merchant Category Recommendation	link	2019.2

https://github.com/microsoft/LightGBM/blob/master/examples/README.md#machine-learning-challenge-winning-solutions

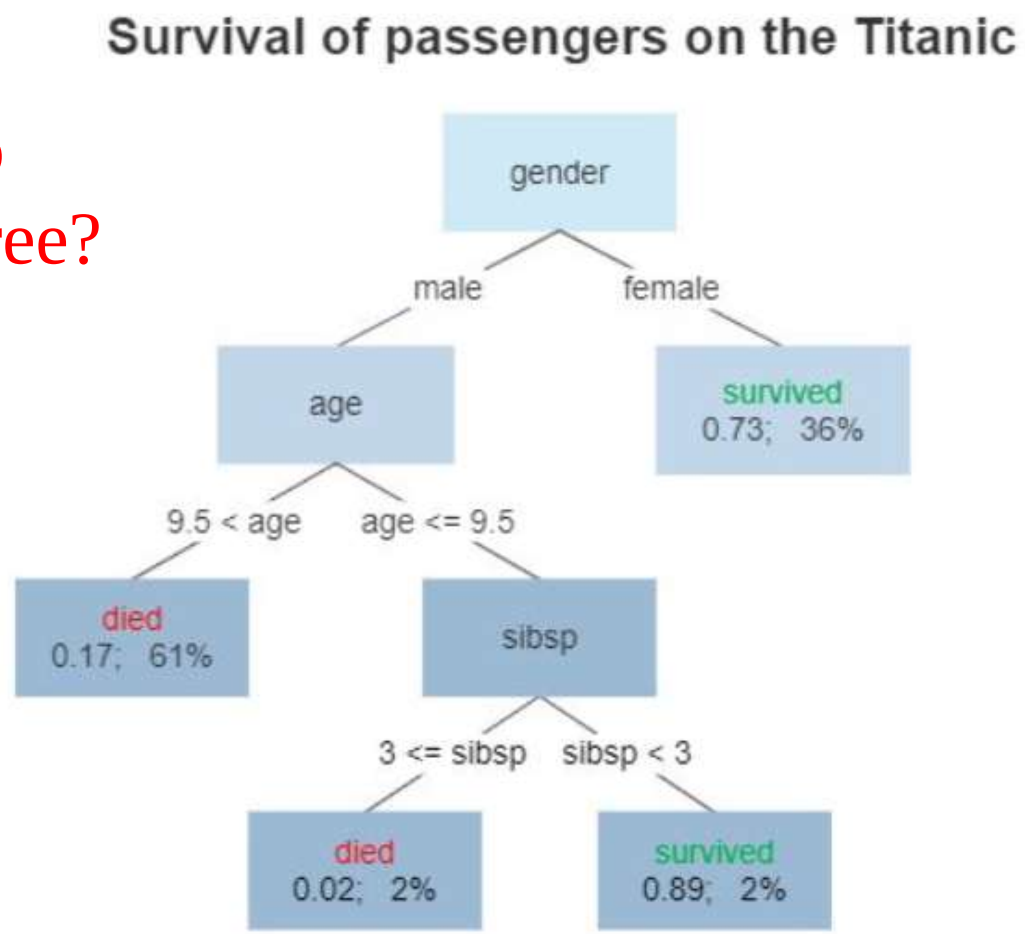
Decision Trees



https://en.wikipedia.org/wiki/Decision_tree_learning#/media/File:Decision_Tree.jpg

Decision Trees

Given a dataset, how to find the best decision tree?



https://en.wikipedia.org/wiki/Decision_tree_learning#/media/File:Decision_Tree.jpg

Tree Split Criteria

- Estimate of Positive Correctness

$$E_P = TP - FP$$

- Gini impurity

$$\mathrm{I}_G(p) = \sum_{i=1}^{J} \left(p_i \sum_{k \neq i} p_k \right) = \sum_{i=1}^{J} p_i (1 - p_i) = \sum_{i=1}^{J} (p_i - p_i^2) = \sum_{i=1}^{J} p_i - \sum_{i=1}^{J} p_i^2 = 1 - \sum_{i=1}^{J} p_i^2$$

- MART gain

$$\frac{1}{s} \left[\sum_{i=1}^{s} (r_{i,k} - p_{i,k}) \right]^2 + \frac{1}{N-s} \left[\sum_{i=s+1}^{N} (r_{i,k} - p_{i,k}) \right]^2 - \frac{1}{N} \left[\sum_{i=1}^{N} (r_{i,k} - p_{i,k}) \right]^2$$

$r_{i,k} = 1$ if $y_i = k$ and $r_{i,k} = 0$ otherwise

$p_{i,k} = \mathbf{Pr}\left(y_i = k | \mathbf{x}_i \right)$

Decision Trees

- Bagging
- Random Forest
- Gradient Boosting

Gradient Boosting

$$p_{i,k} = \mathbf{Pr}\left(y_i = k|\mathbf{x}_i\right) = \frac{e^{F_{i,k}(\mathbf{x_i})}}{\sum_{s=1}^{K} e^{F_{i,s}(\mathbf{x_i})}}, \qquad i = 1, 2, ..., N,$$

where $F_{i,k}(\mathbf{x}_i)$ is an additive model of M terms: $F^{(M)}(\mathbf{x}) = \sum_{m=1}^{M} \rho_m h(\mathbf{x}; \mathbf{a}_m),$

$$L = \sum_{i=1}^{N} L_i, \qquad L_i = -\sum_{k=1}^{K} r_{i,k} \log p_{i,k}$$

where $r_{i.k} = 1$ if $y_i = k$ and $r_{i.k} = 0$ otherwise.

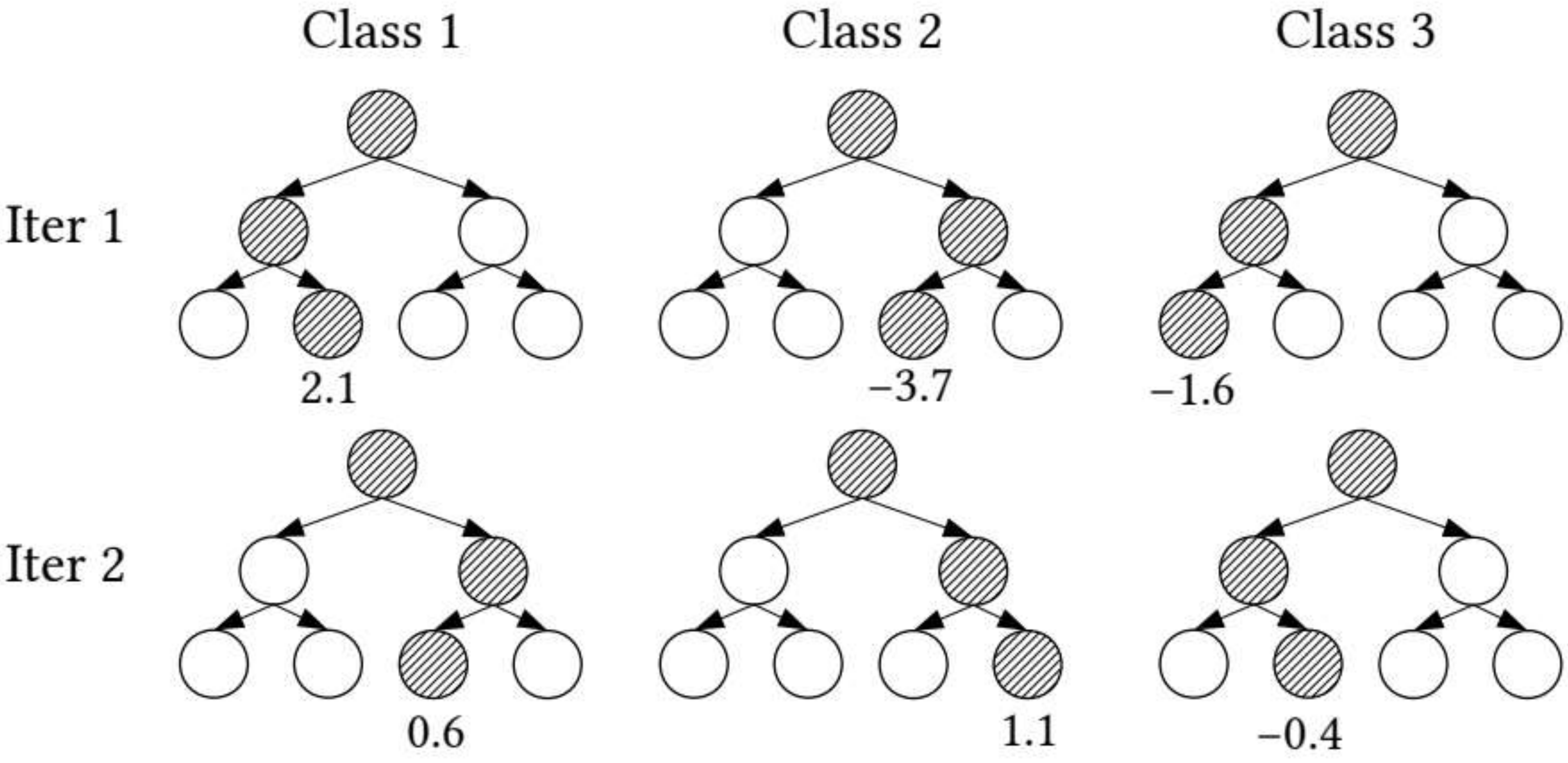
$$\frac{\partial L_i}{\partial F_{i,k}} = -\left(r_{i,k} - p_{i,k}\right), \qquad \frac{\partial^2 L_i}{\partial F_{i,k}^2} = p_{i,k}\left(1 - p_{i,k}\right).$$

Gradient Boosting

Algorithm 1 Robust LogitBoost. MART is similar, with the only difference in Line 4.

1: $F_{i,k} = 0$, $p_{i,k} = \frac{1}{K}$, $k = 1$ to K, $i = 1$ to N
2: **for** $m = 1$ to M **do**
3:     **for** $k = 1$ to K **do**
4:         $\{R_{j,k,m}\}_{j=1}^{J}$ = J-terminal node regression tree from $\{r_{i,k} - p_{i,k},\ \ \mathbf{x}_i\}_{i=1}^{N}$, with weights $p_{i,k}(1 - p_{i,k})$, using the tree split gain formula
5:         $\beta_{j,k,m} = \frac{K-1}{K} \frac{\sum_{\mathbf{x}_i \in R_{j,k,m}} r_{i,k} - p_{i,k}}{\sum_{\mathbf{x}_i \in R_{j,k,m}} (1 - p_{i,k}) p_{i,k}}$
6:         $f_{i,k} = \sum_{j=1}^{J} \beta_{j,k,m} 1_{\mathbf{x}_i \in R_{j,k,m}}$, $\qquad F_{i,k} = F_{i,k} + \nu f_{i,k}$
7:     **end for**
8:     $p_{i,k} = \exp(F_{i,k}) / \sum_{s=1}^{K} \exp(F_{i,s})$
9: **end for**

Gradient Boosting



$$F_{2,1} = 2.1 + 0.6 = 2.7 \qquad F_{2,2} = -3.7 + 1.1 = -2.6 \qquad F_{2,3} = -1.6 - 0.4 = -2.0$$

$$p_{2,1} = 0.986 \qquad p_{2,2} = 0.005 \qquad p_{2,3} = 0.009$$

Data Reading

- Matrix Format
- CSV
- LIBSVM