Click Prediction Boosting via Ensemble Learning Pipelines

Çağatay Demirel¹, ², A. Aylin Tokuç³, and Ahmet Tezcan Tekin¹

¹ Computer Engineering Department, Istanbul Technical University, Istanbul, Turkey
² Donders Institute for Brain, Cognition and Behaviour, Radboud University Medical Center, Nijmegen, Netherlands
³ Computer Engineering Department, Kadir Has University, Istanbul, Turkey

Abstract. Online travel agencies (OTA’s) advertise their website offers on meta-search bidding engines. The problem of predicting the number of clicks a hotel would receive for a given bid amount is an important step in the management of an OTA’s advertisement campaign on a meta-search engine, because \( \text{bid} \times \text{number of clicks} \) defines the cost to be generated. Various regressors are ensembled in this work to improve click prediction performance. Following the preprocessing procedures, the feature set is divided into train and test groups depending on the samples’ logging dates. The data collection is then subjected to XGBoost-based dimension reduction, which significantly reduces the dimension of features. The optimum hyper-parameters are then found by applying Bayesian Hyper-parameter optimization to the XGBoost, LightGBM, and SGD models. Individually, ten distinct machine learning models are tested, as well as combining them to create ensemble models. Three alternative ensemble solutions have been suggested. The same test set is used to test both individual and ensemble models, and the results of 46 model combinations demonstrate that stack ensemble models yield the desired R² score of all. In conclusion, the ensemble model improves the prediction performance by about 10%.

Keywords: Machine Learning · Ensemble Learning · Stack Ensemble · Blend Ensemble · Feature Reduction · Bayesian Optimization · Meta-search Engines.

1 Introduction

Millions of travelers book hotel accommodation over the Internet each year. Modern travelers rely on peer options, electronic word of mouth (eWOM), and peer reviews. Popular online travel websites offer reliable reviews and prices [1]. Therefore, customers choose to inspect and compare different options on meta-search sites like Kayak.com, Trivago, and TripAdvisor before booking their accommodations.

Online travel agencies (OTA’s) advertise their website offers on meta-search bidding engines. If the OTA chooses to have a Cost-Per-Click (CPC) ad campaign,
the OTA promises to pay a certain amount for each click a certain hotel gets from the platform under predefined conditions. The amount to pay per click is the OTA’s bid amount. The problem of predicting the number of clicks a hotel would get for a certain bid amount is an important step in the OTA’s advertisement campaign management on a meta-search engine, as \( \text{bid} \times \text{number of clicks} \) defines the cost to be generated.

In one study, state-of-the-art prediction algorithms and Extreme Gradient Boosting (XGBoost) [2] regressor as well as a minimum Redundancy-Maximum Relevance (mRMR) [3] feature selection algorithm were executed to predict the daily clicks to be received per hotel, using a large OTA’s data from Turkey [4]. The data set received from the meta-search bidding engine contained both numerical and categorical features, with each column having missing and outlier values. The number of clicks as the multiplication of the predicted click-through rate (CTR) and the predicted hotel impression were modelled. The highest R-Squared values obtained in the prediction of individual-hotel based CTR and impression values are both achieved by XGBoost in this work.

Another study aimed to forecast how many impressions and clicks a hotel will acquire as well as how many rooms it will sell via a meta-search bidding engine [5]. The given model predicts how much money an OTA’s hotels will make the following day. The authors demonstrate that by incorporating OTA-specific information into prediction models, the generalization of models improves and better results are obtained. In that study, the best results were obtained using tree-based boosting techniques.

Predicting hotel searches, clicks, and bookings is a challenging task due to many external factors such as seasonality, events, location, and hotel-based properties. Capturing such properties increases the accuracy of prediction models. Due to the high variance in daily OTA data, the use of non-linear prediction methods and creating relevant features with a time-delayed data preprocessing approach is adopted in a work trying to forecast daily room sales for each hotel in a meta-search bidding platform [6]. They applied XGBoost, random forest, gradient boosting, deep neural networks, and generalized linear models (GLM) [7]. The most successful model to predict bookings is gradient boosting, applied on a dataset enriched by features that can summarize the trends in the target variable well.

The demand for hotel rooms in the hotel industry in Turkey between the years 2002-2013 is estimated using ARIMA by Efendioglu and Bulkan [8]. In their study, they determined the hotel room capacity according to the cost of the unsold rooms and the ARIMA distribution. They also reported that the hotel room demand in the country could be affected by outer factors such as political crises and warnings about terrorism. This work shows the non-deterministic nature of hotel room demand and how unpredictable factors suddenly affect the click prediction problem.
In the literature, studies are focusing on the problem of predicting the CTR of a sponsored display advertisement to be shown on a search engine, related to a query. Click and CTR prediction is an ongoing research for both industry and academia [9] [10] [11]. Our aim of predicting the number of clicks is highly related to the CTR prediction problem, hence those studies are investigated to get a better understanding of related work.

In order to predict ad clicks, Google makes use of logistic regression with improvements in the context of traditional supervised learning based on an FTRL-Proximal online learning algorithm [12] for better sparsity and convergence. Microsoft’s Bing Search Engine proposes a new Bayesian online learning algorithm for CTR prediction for sponsored search [13], which is based on a probit regression model that maps discrete or real-valued input features to probabilities. The scalability of the algorithm is ensured through a principled weight pruning procedure and an approximate parallel implementation. Yahoo adopts a machine learning framework based on Bayesian logistic regression to predict click-through and conversion rates [14], which is simple, scalable, and efficient. Facebook combines decision trees with logistic regression [15], generating 3% better results in click prediction, compared to other methods.

Ensemble learning [16] is a machine learning model combination that gets decisions from various models to enhance the overall performance. The ensemble approach provides the stability and low-variety predictions of machine learning algorithms. It builds a set of decision-makers, namely classifiers and regressors, with various techniques as final decisions [17].

An ensemble model is proposed by Wang et al. to predict the CTR of advertisements on search engines [18]. Firstly, they tried several Maximum Likelihood Estimation (MLE)-based methods to exploit the training set; including Online Bayesian Probit Regression (BPR) [19], Support Vector Machine (SVM), and Latent Factor Model (LFM) [20] and optimized them by selecting the most descriptive features. They have created a rank-based ensemble model using the outputs of BPR, SVM, and MLE. The results are ensembled using harmonic means to generate the final blending submission. The proposed model’s output shows an on average 0.013 improvement over the individual models.

In 2015, King et al. investigated whether they could increase the profitability of pay-per-click (PPC) campaigns by using ensemble learning techniques [21]. They applied voting, bootStrap aggregation (Bagging) [22], stacked generalization (or stacking) [23], and metacost [24] techniques to four base classifiers, namely, Naïve Bayes, logistic regression, decision trees, and Support Vector Machines. The research in this work analyzed a data set of PPC advertisements placed on the Google search engine, aiming to classify PPC campaign success. They used average accuracy, recall, and precision metrics to measure the performance of both base classifiers and ensemble models. They also introduced the evaluation metric of total campaign portfolio profit and illustrated how relying on overall model accuracy can be misleading. They conclude that applying
ensemble learning techniques in PPC marketing campaigns can achieve higher profits.

Ling et al. proposed 8 ensemble methods to accurately estimate the CTR in sponsored search ads [25]. A single model would lead to sub-optimal accuracy, and the regression models all have different advantages and disadvantages. The ensemble models are created via bagging, boosting, stacking, and cascading. The training data is collected from historical ads’ impressions and the corresponding clicks. The Area under the Receiver Operating Characteristic Curve (AUC) and Relative Information Gain (RIG) metrics are computed against the testing data to evaluate prediction accuracy. They conclude that boosting is better than cascading for the given problem. Boosting neural networks with gradient boosting decision trees turned out to be the best model in the given setting. They conclude that the model ensemble is a promising direction for CTR prediction; meanwhile, domain knowledge is also essential in the ensemble design.

Etsy, an online e-commerce platform, displays promoted search results, which are similar to sponsored search results and our problem with meta-search bidding engines. CTR prediction is utilized in the system to determine the ranking of the ads [26]. They found out that different features capture different aspects, so they classified the features as being historical and content-based. They train separate CTR prediction models based on historical and content-based features, separately. Then, these individual models are combined with a logistic regression model. They reported AUC, Average Impression Log Loss, and Normalized Cross-Entropy metrics to compare the models to non-trivial baselines on a large-scale real-world dataset from Etsy, demonstrating the effectiveness of the proposed system.

In this study, we utilize ensemble learning algorithms to predict the number of clicks a hotel will receive the next day, and comparing substantial amount of stand-alone models’ prediction performance.

2 Overview of the Proposed System

There are five primary components in the proposed system. The complete system’s flow diagram is depicted in Fig. 1. To summarize, queries are used to retrieve the dataset from the database. Preprocessing is used to extract time-domain seasonal decomposition features with suitable data cleaning in the next stage. XGBoost, LightGBM (LGBM) [27] and Stochastic Gradient Descent (SGD) [28] algorithms are then subjected to hyper-parameter tuning. In the final step, individual and ensembled models are trained and tested with the same train and test sets to generate click predictions. Each model’s R2 score is presented, and 46 distinct models are trained and tested via the proposed system.

2.1 Dataset Generation and Data Preprocessing

The data is retrieved from a major OTA company based on Turkey. Contents of the meta-search platform’s daily reports are combined with the data retrieved
from the OTA. The dataset contains both numerical and categorical features. Some of the columns are eliminated during the data analysis phase as they contain a high ratio of missing values. In this study, we have replaced the missing values with the most common value and the average of the related feature for categorical and numerical features, respectively.

In addition to OTA’s data, some external features are added to the dataset in order to explain the state of the economical and seasonal properties of the environment. Some simple external data examples are daily weather information and daily exchange rates. Data enrichment improves the quality of the dataset. The closeness of the related day to the next public holiday and the length of the holiday are also added as additional numerical variables.

In order to improve the accuracy and generalization ability of the prediction model, additional features are generated from the data following a sliding-window (time-delay) approach. For example, the average and standard deviation of numerical values for some specific time periods (such as the last 3, 7, and 30 days) are calculated and used as input features for prediction. The aim of adding such features is to improve the accuracy and generalization ability of the prediction model.

Feature space is enriched with the seasonal decomposition of some time-series features. Seasonal decomposition is a naive decomposition model that generates
additive components by breaking the original feature into 3. The output of the algorithm is $T$: Trend, $S$: Seasonality, and $e$: Residual, where $Y[t] = T[t] + S[t] + e[t]$. The seasonal component is first removed by applying a convolution filter to the data. The average of this smoothed series for each period is the returned seasonal component \[29\]. Decomposed seasonality, trend, and residual values are added to the dataset as new features.

As a final step, feature one-hot encoding is proposed for some of the string-based features and binarized. In the last step, the feature set is normalized with min-max scaling to force values to be between 0 and 1.

2.2 Bayesian Hyper-parameter Optimization

Hyper-parameter optimization is an essential approach for some machine learning models to enhance prediction performance. There are a few algorithms for tuning hyper-parameters. One of them is a Grid Search \[30\] which tries each combination of given hyper-parameter candidates of a model. Another optimization algorithm is known as random search \[31\], which randomly extracts hyper-parameter combinations and tries to reach local optima of a performance score. However, none of them are able to reach successful local optima of performance in a short period. Bayesian hyper-parameter optimization \[32\] is a relatively more powerful and efficient algorithm for hyper-parameter tuning. It aims to reach a global optimum in a much shorter time than grid search. There is a probabilistic model of $f(x)$ that aims to be exploited to make decisions about where $X$ is accepted as the next performing function. This procedure helps to find the minimum of non-convex functions in a few epochs, which positively effects the performance. The evaluation metric to rank hyper-parameter combinations through input data is R-squared (R²). R-squared is a statistical measure that represents the proportion of the variance for a dependent variable that’s explained by an independent variable or variables in a regression model. The formula of R² is shown in Eq. 1.

$$ R^2 = 1 - \frac{Explained\ Variation}{Total\ Variation} $$

In this work, R² values of individual machine learning algorithms (XGBoost, LightGBM, SGD, Lasso \[33\], Lasso Lars \[34\], Ridge \[35\], Bayesian Ridge \[36\], Huber \[37\], Passive Aggressive Regressors \[38\] and Elastic Net \[39\]) are used and compared in ensemble models.

2.3 Ensembling

If there are $M$ models with errors extracted from the same dataset which are uncorrelated with them, the average error of a model is theoretically reduced by some factor by simply averaging the model outputs. On the other hand, if some of the model outputs have lower performance and are not fit to predict results as well as others, overall error may not be reduced or even increase in some cases.
Average & Weighted Average of Model Outputs

The first and most basic ensembling approach is to take an average of various model outputs. There are two different averaging techniques for ensembling. The first one is taking a mean of predicted values. It provides a lower variance of predicted values since different algorithms proceed to predict various aspects of the input data set. The formula for an average of model outputs is shown in Eq. 2.

$$\text{Avg}_i = \frac{\sum_{r} p_{ir}}{n}$$

where $i$ is the $i^{th}$ sample, $r$ is regressor model, $p_{ir}$ is individual probability of given regressor and $n$ is the number of models used.

However, some machine learning models perform worse than others in terms of prediction, culminating in a poorer overall ensemble prediction performance than some individual regressor prediction performances. The fundamental reason for this is because we give weak regressors the same weight as other regressors that provide decent individual performance. As a consequence, while taking an average of all forecasts, the weighted average is also utilized in this study to eliminate the detrimental influence of low-performance models. Weights are produced using each model’s individual R2 score and scaled between 0 and 1 to standardize the weight of each classifier, ensuring that the sum of all weights is 1. This method allows models that predict higher performance to have a greater impact on final prediction than models that predict lower performance. The formula of the weighted average of model outputs is shown in Eq. 3.

$$\text{Wavg}_i = \sum_r w_r * p_{ir}, r \in R \text{ for } i = 1 \text{ to } N \sum_r w_r = 1$$

where $r$ is the chosen regressor model, $w_r$ is normalized individual R2 performance of regressor, $r$, $p_{ir}$ is prediction result of regressor $r$ of $i^{th}$ sample and $N$ is the number of models used.

Stack Ensemble Model

Stack Ensemble algorithm, assemble results of individual results for different models to make an intermediate input dataset, and the final model is used to create a final regression result. In the proposed approach, 10 different models (XGBoost, LGBM, SGD, Lasso, Lasso Lars, Ridge, Bayesian Ridge, Huber, Passive Aggressive Regressors, and Elastic Net) are trained to stack their extracted predictions, and the intermediate dataset, which is the input to ensemble regressors, is also trained with 6 different models individually to extract 6 different R2 results.

Stacking the individual predictions enables us to analyze the intermediate regressor model to linearly weight results to create a learnable weighted average of provided predictions through each sample of input data. The stack ensemble model flow diagram is shown in Fig. 2.
Fig. 2. Stack Ensemble Model
**Blend Ensemble Model** Blend ensemble algorithm [40] is designed similarly to the stack ensemble algorithm. In the first step, individual results of regressor models are assembled. In addition to this, the original feature set allowed leak into stack predictions to combine with them, which provides an increased feature dimension that has the knowledge of desired predictions.

Similar to stack ensemble models, XGBoost, LightGBM, SGD, Lasso, Lasso Lars, Ridge, Bayesian Ridge, Huber, Passive Aggressive Regressors, and Elastic Net are used to stack their given prediction outputs and blended with the input feature set. Then, the blended dataset is also trained with four different models individually to extract four different R2 results. The blend ensemble model flow diagram is shown in Fig. 3.

### 3 Experiments and Results

#### 3.1 Click prediction Results of Various Models

Instead of splitting a dataset into train and test with some percentage, daily click predictions of each hotel are estimated. Accordingly, the train set is designed from the earliest day until test day that clicks will be predicted. By using this approach, 11 consecutive days are chosen as test days and 11 corresponding R2 test scores are produced by processing 3 different ensembling models (Average & weighted average, stack ensemble, and blend ensemble). Besides, individual R2 test scores of 10 regressor models (XGBoost, LightGBM, SGD, Lasso, Lasso Lars, Ridge, Bayesian Ridge, Huber, Passive Aggressive Regressors, and Elastic Net) are reported for the control group, and efficiency of ensemble models is evaluated.

For each test day, 22 different predictions are measured (10 individual predictions, average & weighted average predictions, 5 stack ensemble prediction, and 4 blend ensemble predictions). R2 score of each prediction is saved and the average of each test R2 score is calculated. The average R2 test scores of 21 model types are shown in Fig. 4.

### 4 Conclusion

Various regressors are ensembled in the proposed study to improve click prediction performance. The feature set is divided into train and test groups depending on the logging date in the first phase. The data collection is then subjected to an XGBoost-based dimension reduction, which significantly reduces the dimension of features. To discover the most ideal hyper-parameters, Bayesian Hyperparameter optimization is developed for the XGBoost, LightGBM, and SGD models. XGBoost, LightGBM, SGD, Lasso, Lasso Lars, Ridge, Bayesian Ridge, Huber, Elastic Net, and Passive Aggressive regressors are all tested separately and then fused to create ensemble models.
Fig. 3. Blend Ensemble Model
Click Prediction Boosting via Ensemble Learning Pipelines

![Average R2 Scores](image)

**Fig. 4.** Overall Test R2 Scores for Each Regressor Model
The authors suggest three different ensemble approaches. The first ensemble model takes an average of anticipated results as well as a weighted average. A stack ensemble model, for example, assembles all the results of individual forecasts as an intermediate layer that feeds into another individual layer. The third model is a blend ensemble model, which stacks all of the individual prediction outputs and blends them with the original feature set once more. With the outcomes of multiple model outputs, this framework offers an artificial feature generation to boost the feature dimension.

The same test set is used to test both individual and ensemble models, and the results of 46 model combinations demonstrate that stack ensemble models produce the best R2 score of all. The greatest R2 score is 0.639 for the stack ensemble model combined with linear regression, whereas the best machine learning model had an R2 score of 0.579. As a conclusion, the ensemble model improves prediction performance by about 10%.

Various types of artificial neural network (ANN) models will be added to ensemble models in the future, with the goal of improving stack and blend ensemble models. Yandex’s CatBoost machine learning model [41], which handles categorical information, can also be added to the list of regressors to examine.
References

[1] Luis V Casalo, Carlos Flavian, Miguel Guinaliu, and Yuksel Ekinci. Do online hotel rating schemes influence booking behaviors? *International Journal of Hospitality Management*, 49:28–36, 2015.

[2] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. https://doi.org/10.1145/2939672.2939785 URL http://doi.acm.org/10.1145/2939672.2939785.

[3] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(09):1226–1238, sep 2002. ISSN 1939-3539. https://doi.org/10.1109/TPAMI.2002.1033214.

[4] Tülin Cakmak, Ahmet T Tekin, Cagla Senel, Tugba Coban, Zeynep Eda Uran, and C Okan Sakar. Accurate prediction of advertisement clicks based on impression and click-through rate using extreme gradient boosting. In *Proceedings of the ICPRAM 2019 - 8th International Conference on Pattern Recognition Applications and Method*, 2019.

[5] Ahmet Tezcan Tekin and Ferhan Cebi. Click and sales prediction for digital advertisements: Real world application for otas. In *International Conference on Intelligent and Fuzzy Systems*, pages 205–212. Springer, 2019.

[6] Gizem Aras, Gülsah Ayhan, Mehmet Ali Sarikaya, A Aylin Tokuç, and C Okan Sakar. Forecasting hotel room sales within online travel agencies by combining multiple feature sets. In *Proceedings of the ICPRAM 2019 - 8th International Conference on Pattern Recognition Applications and Method*, 2019.

[7] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972. https://doi.org/https://doi.org/10.2307/2344614 URL https://rss.onlinelibrary.wiley.com/doi/abs/10.2307/2344614.

[8] Deniz Efendioğlu and Serol Bulkan. Capacity management in hotel industry for turkey. In *Handbook of Research on Holistic Optimization Techniques in the Hospitality, Tourism, and Travel Industry*, pages 286–304. IGI Global, 2017.

[9] Daniel C Fain and Jan O Pedersen. Sponsored search: A brief history. *Bulletin of the american Society for Information Science and technology*, 32(2):12–13, 2006.

[10] Bernard J Jansen and Tracy Mullen. Sponsored search: an overview of the concept, history, and technology. *International Journal of Electronic Business*, 6(2):114–131, 2008.
[11] Anindya Ghose and Sha Yang. An empirical analysis of search engine advertising: Sponsored search in electronic markets. Management science, 55 (10):1605–1622, 2009.
[12] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1222–1230, 2013.
[13] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft’s bing search engine. Omnipress, 2010.
[14] Olivier Chapelle, Eren Manavoglu, and Romer Rosales. Simple and scalable response prediction for display advertising. ACM Transactions on Intelligent Systems and Technology (TIST), 5(4):1–34, 2014.
[15] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, pages 1–9, 2014.
[16] S. Lei, M. Xinming, X. Lei, and H. Xiaohong. Financial data mining based on support vector machines and ensemble learning. In 2010 International Conference on Intelligent Computation Technology and Automation, pages 313–314, May 2010. https://doi.org/10.1109/ICICTA.2010.787.
[17] Thomas G. Dietterich. Ensemble methods in machine learning. In Multiple Classifier Systems, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-45014-6.
[18] Xingxing Wang, Shijie Lin, Dongying Kong, Liheng Xu, Qiang Yan, Siwei Lai, Liang Wu, Alvin Chin, Guibo Zhu, Heng Gao, et al. Click-through prediction for sponsored search advertising with hybrid models. In KDD Workshop, 2012.
[19] Tony E Smith and James P LeSage. A bayesian probit model with spatial dependencies. In Spatial and spatiotemporal econometrics. Emerald Group Publishing Limited, 2004.
[20] Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 19–28, 2009.
[21] Michael A King, Alan S Abrahams, and Cliff T Ragsdale. Ensemble learning methods for pay-per-click campaign management. Expert Systems with Applications, 42(10):4818–4829, 2015.
[22] Leo Breiman. Bagging predictors. Machine learning, 24(2):123–140, 1996.
[23] S. Zirpe and B. Joglekar. Negation handling using stacking ensemble method. In 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), pages 1–5, Aug 2017. https://doi.org/10.1109/ICCUBEA.2017.8463946.
[24] Pedro Domingos. Metacost: A general method for making classifiers cost-
sensitive. In Proceedings of the fifth ACM SIGKDD international conference
on Knowledge discovery and data mining, pages 155–164, 1999.
[25] Xiaoliang Ling, Weiwei Deng, Chen Gu, Hucheng Zhou, Cui Li, and Feng
Sun. Model ensemble for click prediction in bing search ads. In Proceedings
of the 26th International Conference on World Wide Web Companion, pages
689–698, 2017.
[26] Kamelia Aryafar, Devin Guillory, and Liangjie Hong. An ensemble-based
approach to click-through rate prediction for promoted listings at etsy. In Proceed-
ings of the ADKDD’17, pages 1–6. 2017.
[27] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient
gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio,
H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in
Neural Information Processing Systems 30, pages 3146–3154. Curran Associates,
Inc., 2017. URL http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-
boosting-decision-tree.pdf.
[28] Léon Bottou. Large-scale machine learning with stochastic gradient descent.
In Yves Lechevallier and Gilbert Saporta, editors, Proceedings of COMP-
STAT’2010, pages 177–186, Heidelberg, 2010. Physica-Verlag HD. ISBN
978-3-7908-2604-3.
[29] Nurilla Avazov, Jiamou Liu, and Bakhadyr Khoussainov. Periodic neural
networks for multivariate time series analysis and forecasting. In 2019 Inter-
national Joint Conference on Neural Networks (IJCNN), pages 1–8, 2019.
https://doi.org/10.1109/IJCNN.2019.8851710.
[30] James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Al-
gorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. S.
Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, Advances in
Neural Information Processing Systems 24, pages 2546–2554. Curran Associates,
Inc., 2011. URL http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-
optimization.pdf.
[31] Dean C. Karnopp. Random search techniques for optimization
problems. Automatica, 1(2):111 – 121, 1963. ISSN 0005-
1098. https://doi.org/https://doi.org/10.1016/0005-1098(63)90018-9.
[9] URL http://www.sciencedirect.com/science/article/pii/
0005109863900189.
[32] V. Nguyen. Bayesian optimization for accelerating hyper-parameter tun-
ing. In 2019 IEEE Second International Conference on Artificial Intelli-
gen and Knowledge Engineering (AIKE), pages 302–305, June 2019.
https://doi.org/10.1109/AIKE.2019.00060.
[33] Robert Tibshirani. Regression shrinkage and selection via the lasso.
Journal of the Royal Statistical Society: Series B (Methodological), 58
(1):267–288, 1996. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.
URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.
2517-6161.1996.tb02080.x.
[34] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 04 2004. https://doi.org/10.1214/009053604000000067 URL https://doi.org/10.1214/009053604000000067.

[35] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. https://doi.org/10.1080/00401706.1970.10488634 URL https://amstat.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634.

[36] Qi Shi, Mohamed Abdel-Aty, and Jaeyoung Lee. A bayesian ridge regression analysis of congestion’s impact on urban expressway safety. *Accident Analysis & Prevention*, 88:124 – 137, 2016. ISSN 0001-4575. https://doi.org/https://doi.org/10.1016/j.aap.2015.12.001 URL http://www.sciencedirect.com/science/article/pii/S0001457515301524.

[37] Qiang Sun, Wen-Xin Zhou, and Jianqing Fan. Adaptive huber regression. *Journal of the American Statistical Association*, 0(0):1–24, 2019. https://doi.org/10.1080/01621459.2018.1543124 URL https://doi.org/10.1080/01621459.2018.1543124.

[38] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 03 2006.

[39] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. https://doi.org/10.1111/j.1467-9868.2005.00503.x URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00503.x.

[40] Z. Xie, A. Singh, J. Uang, K. S. Narayan, and P. Abbeel. Multimodal blending for high-accuracy instance recognition. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2214–2221, Nov 2013. https://doi.org/10.1109/IROS.2013.6696666.

[41] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobjev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: Unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS’18, page 6639–6649, Red Hook, NY, USA, 2018. Curran Associates Inc.