Copyright ⓒ 2023 KSII

Machine Learning vs. Deep Learning on a Tabular Traffic Dataset in Big Data

Dalyapraz Manatova, Jongwook Woo
Luddy School of Informatics, Indiana University, USA
Department of Information Systems, California State University Los Angeles, USA
[e-mail: dmanato@iu.edu, jwoo5@calstatela.edu]
*Corresponding author: Jongwook Woo

Abstract

This paper compares the performance of traffic prediction models in machine learning and deep learning capable of classifying a multiclass traffic dataset. The dataset comprises five different traffic jam levels of tabular format. Its data size exceeds 1.8 GB, making storing and processing using conventional single-node machines impractical. To address this issue, we developed classification models on a Big Data platform using machine learning algorithms, specifically Random Forest and XGBoost. Additionally, we implemented a deep learning model on the platform using feed-forward multilayer artificial neural network. Then, we evaluated the performance of the models in terms of accuracy and computing time. The data is in a tabular format, and the deep learning model did not perform better in the predictive analysis than the machine learning models. The Random Forest model presented the highest accuracy and efficiency in computing time. Our experimental results indicate that the deep neural network model was less effective than tree-based machine learning models in predicting tabular format multiclass traffic data.

Keywords: Spark, Big Data, GPU, Rapids, Deep Learning, Tabular Data, Scalable Computing

1. Introduction

Deep Learning algorithms are known as superior to Machine Learning algorithms, especially with large-scale datasets. However, some show it is not always true [2, 3].

Spark platforms is linearly scalable as distributed parallel computing systems that store and process large-scale datasets. It also supports multiclass classification algorithms: logistic regression, decision trees, random forests, and naive Bayes. In this paper, we compare the models to predict traffic jams using a traffic dataset collected from smartphone device apps. The dataset is over 1.8 GB in a tabular format. The dataset has five-level traffic jams, which requires a multiclass classification model. We also study if the Deep Learning (DL) model could perform better than the Machine Learning (ML) model with tabular datasets [2, 3].

2. Related Work

Dalya et al. improved the performance of machine learning in distributed parallel computing systems with GPU to predict binary-class traffic jam [1]. In this paper, we develop multiclass traffic jam prediction models in Big Data systems, including a DL model.

Ravid et al. compared the accuracy of an XGBoost and various Deep Learning models with 11 tabular data sets. It showed that XGBoost model performs better than deep learning models [2]. Léo et al. investigated the performance of Deep Learning and Tree-based models with 24 tabular datasets. They found that tree ensemble models are superior to deep learning. They stated that it is because the tabular dataset has irregular patterns in the target function, uninformative features, and non-rotationally-invariant data [3]. Our paper's models are multiclass classification in the big

This research was supported by a research credit from Amazon AWS and Oracle Cloud

data platform with over 1.8 GB size, while they compared binary classification and regression models with small data sets.

3. Big Data Machine Learning and Deep Learning with GPU

Apache Spark has been a popular Big Data solution. GPU chip uses multi-cores for parallel computing to achieve high performance and accelerates the development of the deep learning applications. The Rapids suite is an open-source software libraries to utilize GPU [5]. We can leverage Spark Big Data cluster with Rapids in multiple nodes that have both CPUs and GPU.

4. Dataset and Specifications

We collaborated with the Information Technology Agency of the Los Angeles City Department with the traffic dataset. The dataset comprises JSON files covering information reported by app users and information captured from users' smartphones. The dataset is of size 1.8 GB in Dec 31, 2017 – Jan 8, 2018. The data has two significant files: alerts (information reported by users) and jams (information captured by the user's device). We store the dataset in the Amazon cluster, EMR 6.8, which is consisted of 3 nodes: one *m3.xlarge* and two *g4dn.xlarge*. Each node has 4 cores and 16 GB memory. It supports Spark and Hadoop File Systems. The dataset comprises columns from the device and the app users: *[location_x, location_y, address, road_type, type, pub_date, date_pst, month, day, hour, min, sec, weekday, speed, length, delay, report_description, level]*. The column, *level*, is the multiclass label showing traffic jam level. It ranged from 1 to 5, where 1 is "almost no jam" and 5 is "standstill jam". We build machine learning and deep learning models to predict the *five-level* (*five-classes*) traffic jam.

5. Multiclass Classification Models

We develop multiclass classification models to classify five-level traffic jams: RF (Random Forest), XGBoost, and FF (Feed-Forward Multilayer Neural Network). FF contains many hidden layers with neurons and stochastic gradient descent using back-propagation.

5.1 Data Engineering and Modeling in Machine Learning

We partitioned our dataset into 80% training and 20% testing sets. Besides, we put the hyper-parameters of *RF* model as *numTrees*: [10, 20], *minInfoGain*: [0.0, 0.01], and *maxDepth*: [5, 10]. For *XGBoost*, we set the hyper-parameters as *ntrees*: 50, and *max_depth*: 5. In FF, we put *hidden layer*: (200, 200), and *epochs*:100. We evaluate the performance of the models by employing the metrics, *Precision*. It denotes the ratio of correctly identified positive records out of the total records identified as positive. Another metric is *Recall*, which represents the ratio of correctly identified positive records out of the total actual positives. We also measure Macro- and Micro-Average Precision & Recall, as the label has five classes/levels. We focus on Macro-Average more because it is affected by the majority class.

5.2 Experimental Result

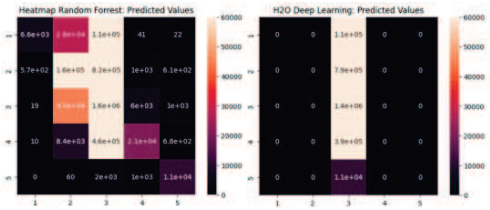


Fig. 1. Confusion Matrix Heat Maps of Random Forest and Feed-Forward Deep Learning

We calculate the metrics of multiclass classification models with a *Confusion Matrix* composed of Actual and Predicted values of the five levels. Its rows are Actual Values, and the columns are Predicted.

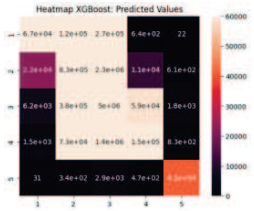


Fig. 2. Confusion Matrix Heat Map of XGBoost

RF of **Fig. 1** shows that classes 2 and 3 have higher TP (True Positive) values than other

classes. It also shows that class 3 of FF has the only highest TP while other predicted valuse are 0s. Figure 2 illustrates that XGBoost has more stable and high TP values than other models. TP is displayed in the diagonal of the matrix. Table 1 shows macro-average Precision & Recall of the models and the computation time to build a model. RF has the shortest computing time. The computation time using GPU in XGBoost is 9 % faster than CPU.

Table 1. Performance Comparison of the Models

Macro -Avg	RF	XGBoost		FF (Epochs)	
		CPU	GPU	50	100
Prec.	0.733	0.690		0	
Recall	0.400	0.468		0.2	
Comp. (mins)	38	66	60	59	120

Table 2. Accuracy of Random Forest Model

	Precision	Recall
Class 1	0.91697	0.04601
Class 2	0.66767	0.16480
Class 3	0.54003	0.96947
Class 4	0.71825	0.04177
Class 5	0.82419	0.77846

Tables 2 and **3** list each Precision & Recall of the five classes in RF and XGBoost models. We focus more on the high Precision in the low classes. Class 1 with the highest Precision is helpful for the drivers, where the higher the Precision of Class 1, the smaller the FP (False Positive) of Class 1 is. We can translate the FP as follows: a model predicts that a place is Class 1 ("almost no jam"), but it is false. So, it has a standstill jam, which will annoy the drivers who follow the prediction. The RF model has higher *Precision* of 91.7 % and 66.8 % in Classes 1 and 2 than the XGBoost model with 69.3 % and 59.3 %. It is challenging to use the FF model that has only a precision of 50.9 % in Class 1, while other classes have Precisions of 0s.

Table 3. Accuracy of XGBoost Model

	Precision	Recall
Class 1	0.69281	0.14494
Class 2	0.59282	0.26392
Class 3	0.55945	0.91817
Class 4	0.67765	0.09465
Class 5	0.92680	0.91651

6. Conclusions

This paper compares the performance of traffic prediction models capable of classifying a multiclass traffic dataset with five traffic jam levels exceeding 1.8 GB in size. We evaluated the performance of the multiclass classification models in terms of accuracy and computing time using ML and DL algorithms on a Big Data Spark platform: RF, XGBoost, and FF. The FF model did not exhibit higher accuracy than the ML models, as the data was in a tabular format. The RF model demonstrated the highest *Precision* (91.7 %) with the most efficient computing time (38 mins), especially in predicting low traffic jam levels 1 and 2. The XGBoost model showed generally higher *Precision* in all traffic jam levels with a 9% faster computation time with GPU acceleration. In contrast, the DL model efficiently predicted only traffic jam level 3. Our experimental results show that tree-ensemble ML models effectively predict multiclass traffic jam levels in tabular data format than a DL FF model.

References

[1] Dalyapraz Dauletbak, Junghoon Heo, Sooyoung Kim, Yeon Pyo Kim and, Jongwook Woo, "Scalable Traffic Predictive Analysis for Smart City using GPU in Big Data," KSII The 16th Asia Pacific International Conference on Information Science and Technology (APIC-IST), pp144-148, ISSN 2093-0542 June 20-22 2021.

[2] Ravid Shwartz-Ziv, and Amitai Armon, "Tabular data: Deep learning is not all you need," Journal of Information Fusion, vol 81, pp. 84-90, 2022, Elsevier

[3] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," Thirty-sixth Conference on Neural Information Processing Systems (neurIPS) Dec 2-4 2022.

[4] M. Schnuerle, "Louisville and Waze: Applying Mobility Data in Cities," Harvard Civic Analytics Network Summit on Data-Smart Government, 2017.