Predicting the ratings of Amazon products using Big Data

Article *in* Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery · December 2020

DOI: 10.1002/widm.1400

2 authors:

Jongwook Woo
California State University, Los Angeles
40 PUBLICATIONS **232** CITATIONS
SEE PROFILE

Monika Mishra
California State University, Los Angeles
3 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Received: 15 April 2020 | Revised: 4 November 2020 | Accepted: 13 November 2020

DOI: 10.1002/widm.1400

ADVANCED REVIEW

WIREs DATA MINING AND KNOWLEDGE DISCOVERY

WILEY

Predicting the ratings of Amazon products using Big Data

Jongwook Woo | **Monika Mishra**

Department of Information Systems,
California State University Los Angeles,
Los Angeles, California

Correspondence
Jongwook Woo, Department of Information Systems, California State University Los Angeles, Los Angeles, CA.
Email: jwoo5@exchange.calstatela.edu

Funding information
Oracle Cloud Innovation Accelerator

Abstract

This paper aims to apply several machine learning (ML) models to the massive dataset present in the area of e-commerce from Amazon to analyze and predict ratings and to recommend products. For this purpose, we have used both traditional and Big Data algorithms. As the Amazon product review dataset is large, we present Big Data architecture suitable massive dataset for storing and computation, which is not possible with the traditional architecture. Furthermore, the dataset contains 15 attributes and has about 7 million records. With the dataset, we develop several models in Oracle Big Data and Azure Cloud Computing services to predict the review rating and recommendation for the items at Amazon. We present a comparative conclusion in terms of the accuracy as well as the efficiency with Spark ML—the Big Data architecture, and Azure ML—the traditional architecture.

This article is categorized under:

Fundamental Concepts of Data and Knowledge > Big Data Mining
Technologies > Machine Learning
Technologies > Prediction

KEYWORDS

big data, predictive analysis, spark, distributed computing, scalable computing, machine learning

1 | INTRODUCTION

The exponential growth of the Internet and its increasing accessibility have accelerated the development of electronic commerce over the past few years. Electronic Commerce is also known as e-commerce or e-business. It is the purchasing and selling of products or services through electronic systems over the Internet.

E-commerce has various advantages as it overcomes geographic limitations and eliminates travel time and cost. E-commerce allows us to visit the store virtually, with just a few mouse clicks. It is very convenient. An online store is available all day, every day meaning the customers can visit the store at all times, no matter what their schedule might be. E-commerce facilitates comparison shopping. Several online portals allow customers to browse multiple e-commerce sites and find the best-suited prices.

Both businesses and customers have embraced online sales as a convenient way to shop. But E-commerce has disadvantages too. The most significant disadvantage is that one cannot experience the product before purchase. Lack of touch or feel of products during online shopping is a drawback. Because of this constraint, most of the customers, on any online shopping site, make purchasing decisions based on reviews and ratings. Consumers now, more than ever, are looking for ratings and reviews regarding the services and products of companies before making purchasing decisions. Therefore, the business needs to have the insights of items and predict the products' ratings.

In the challenging environment of the electronic marketplace in which competitors are only a click away, web retailers are vulnerable to customer attrition. Customer retention is the most significant factor for business growth. A customer can only retain when he is satisfied. Businesses must be able to stay one step ahead of their customers to keep them in a highly dynamic market. They must be able to predict what customers are looking for in their e-commerce store. Prediction is vital for every business. It gives you an understanding of how a company is going to perform shortly, and accordingly, you can change the various business policies to maximize the profit.

We have used the review dataset to predict customer and product trends, then, to recommend products to the users. However, as there are more than 300 million users and 50 billion items, traditional machine learning (ML) architecture cannot store and process this large-scale dataset, which needs Big Data systems. Woo and Xu (2011) defined Big Data as non-expensive frameworks that can store a vast and variety of dataset and process it as parallel and distributed systems.

Our goal, in this paper, is to predict the rating of products on Amazon in Big Data architecture. The dataset of ratings is of size 3.63 GB with the file format of Tab Separated Values (TSV) and 15 columns. The "Star_rating" column is the label column that has five distinct values—1, 2, 3, 4, and 5. We adopt Spark ML platform as a Big Data solution for predictive computation. Besides, we use Azure ML for the small sample dataset, which is an open tool for traditional predictive analysis. We developed various ML models using Azure ML and Spark ML platforms for the rating prediction. We have also created recommendation models. Recommender systems keep customers on a business site longer, who interact with more products/content, and it suggests products or content to a customer who is likely to purchase. It helps to provide customers personalized shopping experience.

We have also developed text analytics models on the "Review Headline" column to gauge the sentiments of the customers. Just as predictive models analyze past performance to assess how likely a customer is to exhibit a specific behavior in the future to improve marketing effectiveness, text analytics can help companies infer similar insights from analyzing subtle text patterns to answer questions about customer performance.

2 | RELATED WORK

Based on the extensive study reported on various academic papers, we noted some prominent differences with our approach. Patel (2017), Woolf (2017), and Saumya, Singh, Baabdullah, Rana, and Dwivedi (2019) performed an analysis of Amazon product review dataset, but their goals and techniques were quite different from ours.

Patel's (2017) work was to classify Amazon product review into positive and negative. He performed a sentimental analysis for one of the baby products. The tools used in his approach were traditional ML using Python, GraphLab, and S Frame, which is not scalable and should not work for dataset higher than Giga-byte. Patel adopts logistic regression and classification to predict the rating range of 0 to 5 and the binary rating. Our approach is mainly on predictive analysis using the Big Data platform, Spark ML, to predict product ratings, which is linearly scalable for growing datasets. We developed the models of decision tree (DT) regression and gradient boosted regression and logistic regression to predict the rating range of 0 to 5 in PySpark.

Woolf (2017) has done another similar study by performing descriptive analysis using Sparklyr platform that is another Big Data method that is also scalable. The size of the data used in his project was 4.53 GB. In contrast, our research is about predictive analysis. We develop the predictive analysis by implementing various ML models while Woolf's (2017) work was restricted just to descriptive analysis.

Saumya et al. (2019) collected product review data from Amazon India and snapdeal.com, which are total 29,215 and 12,686 review data, respectively, and developed prediction models to find out helpful reviews by classifying the review to high- and low-quality reviews. Three different classification approaches, support vector machine (SVM), naïve Bayes, and random forest, are used for selecting the critical features and classify the reviews, and to predict if the review was helpful using two different regression techniques, linear regression and gradient boosting regression. Saumya's dataset is a few Mega-Byte, and the goal is different from ours because we predict the user's product rating with the larger dataset on the big data platform.

3 | EXPERIMENTAL PLATFORMS AND ARCHITECTURES

We have used three different platforms for the implementation of the ML models: the traditional and Big Data systems. The following describes these platforms.

3.1 | Spark Hadoop Big Data systems and scalability

Apache Spark is a distributed parallel and cluster computing systems and supports MLlib ML APIs. Spark computing engine in the paper uses Hadoop Big Data systems for its Hadoop Distributed File Systems (HDFS) file systems—or Object Storage of Cloud Computing—and YARN resource management, which provides a unified data analytics (UDA) platform. UDA platform is an integrated system for data storage, analysis, and prediction, especially for massive datasets.

Figure 1 shows that the Spark Hadoop cluster can grow by adding more servers as collecting more data. For example, the dataset increases from the 3 GB of data to 200TB of data. Then, the systems require additional storage and computing engines by adding more servers, and it still works well with the TBs of data by well-supported resource management.

Gupta, Le, Boldina, and Woo (2019) and Purushu, Melcher, Bhagwat, and Woo (2018) showed that the Big Data architecture is linearly scalable so that if the architecture can store and compute several Giga-Bytes of data, it should work with hundreds of Giga-Bytes of dataset and more. García-Gil, Ramírez-Gallego, and García (2017) also compare Spark and Flink Big Data platforms with 10, 30, 50, 75, and 100% of the dataset using SVM and Linear Regression models, and shows that both platforms are linearly scalable. They also concluded that Spark ML has better performance than Flink ML.

Therefore, the proposed predictive models of the paper in Spark ML platform should be linearly scalable with more massive datasets if adding more servers.

3.2 | Microsoft Azure ML Studio

Microsoft Azure ML Studio is a collaborative, drag-and-drop tool one can use to build, test, and deploy predictive analytics solutions on the data. The memory available for the platform is 10 GB, and the number of nodes is 1. It is easy and straightforward to use for the relatively small dataset, typically less than 500 MB. Besides, we can find out the models with the small sample dataset using Azure ML Studio, which shows the best performance and used to select Big Data predictive algorithms. We can interactively and quickly deploy Predictive models on Azure ML studio. Azure ML provides clients with black-box modules for each development phase. Besides, it enables model customization by providing Python and R-script Modules. Also, Azure ML provides many ML algorithms, including classification, regression, and clustering (Qasem, Thulasiram, & Thulasiram, 2015).

3.3 | Databricks

We have used the Databricks Community Edition (CE) for implementing Spark ML codes with the sample dataset to develop ML models, which we found from Azure ML. The Databricks CE is the free version of the cloud-based big data platform. Its users can access a micro-cluster as well as a cluster manager and notebook environment.

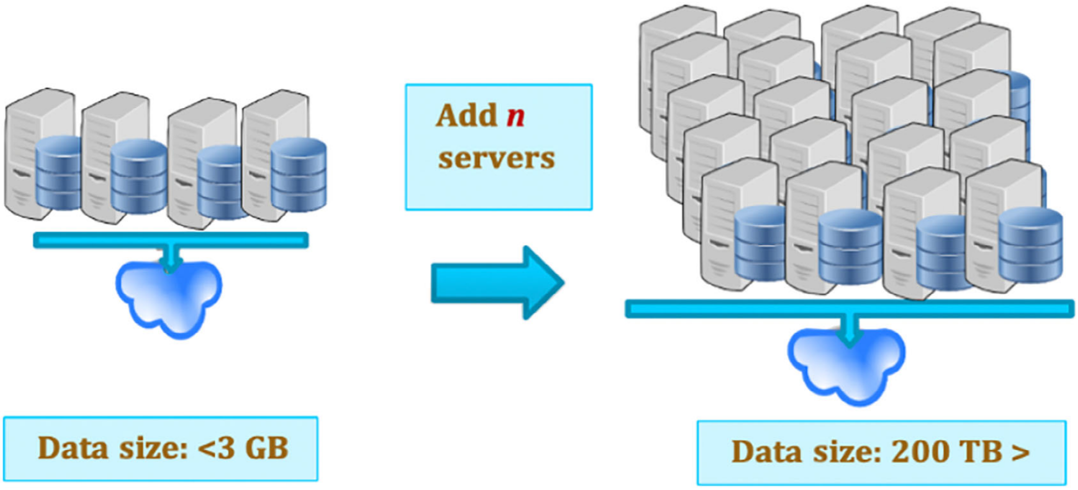

FIGURE 1 Spark Hadoop Big Data UDA platform and the scalability

The Databricks runtime version was 5.3, which included Apache Spark 2.4.0 and Scala 2.11. The Python version used was 3. It had a 6.0 GB memory, 0.88 cores, and 1 DBU.

Databricks provides a high-speed data platform which runs on top of Apache Spark that helps to create Big Data advanced analytics solution easily. It can be connected directly to the existing Amazon storage clusters and Databricks services in the cloud. It provides a highly integrated workspace to create dashboards by using notebooks. Also, it provides functionality to use third-party Business Intelligence tools (Pritwani, Wasley, & Woo, 2018).

3.4 | Oracle BDCE

Azure ML Studio and Databricks CE can store and process small datasets. Because of this file size limit, the ML model could be developed only for a sample data extracted from the full dataset. For a massive dataset—greater than 3 GB, we adopt the Oracle Big Data Compute Edition (BDCE), which is a good fit to perform the Big Data gathering and analysis. Oracle BDCE supports Apache Hadoop with Spark ML. Thus, we launched a Hadoop Spark cluster on the Oracle Big Data Cloud platform for the predictive analysis. The platform is composed of a memory of 180 GB, and the storage capacity was 682 GB. It had an OCPU of 12 with the number of nodes running as 6.

4 | DATASET DETAILS

We acquired the dataset for the paper from the Amazon AWS site ("S3.amazonaws.com", n.d.). The dataset presents the details about the products reviewed on Amazon sites between 2005 and 2015 in the United States. The analyzed Amazon product review dataset contains 15 attributes and has about 6.93 million records. The total file size is 3.63 GB, which is vast so that the traditional systems cannot afford or takes a long time to compute the prediction. The format of the file is TSV.

The list of columns and Table 1 describes the details of the columns. The dataset is of both quantitative and qualitative nature. The dataset size is 3.63 GB, having 6.93 million records, which describes the property of quantitative. The opinions present in *review_headline* and *review_body* columns by the Amazon customers represent the qualitative nature of the dataset. It presents their experiences regarding the products on the Amazon website.

Figure 2 shows the review count as per the product category, which indicates that Video DVD was the maximum reviewed product, which is 1,948,238. The next comes mobile apps with 1,759,144 and then the Digital_ebook_Purchase

TABLE 1 Column name and with column details of the dataset

Column name	Column details
marketplace	Country code—US
customer_id	The ID of the customer
review_id	The unique ID of the review
product_id	The unique product ID the review pertains to
product_parent	Random identifier used to aggregate reviews for same product
product_title	Title of the product
product_category	Broad product category that can be used to group reviews
star_rating	The 1–5 star rating of the review
helpful_votes	Number of helpful votes
total_votes	Number of total votes the review received
vine	Review was written as part of the Vine program
verified_purchase	The review is on a verified purchase
review_headline	The title of the review
review_body	The review text
review_date	The date the review was written

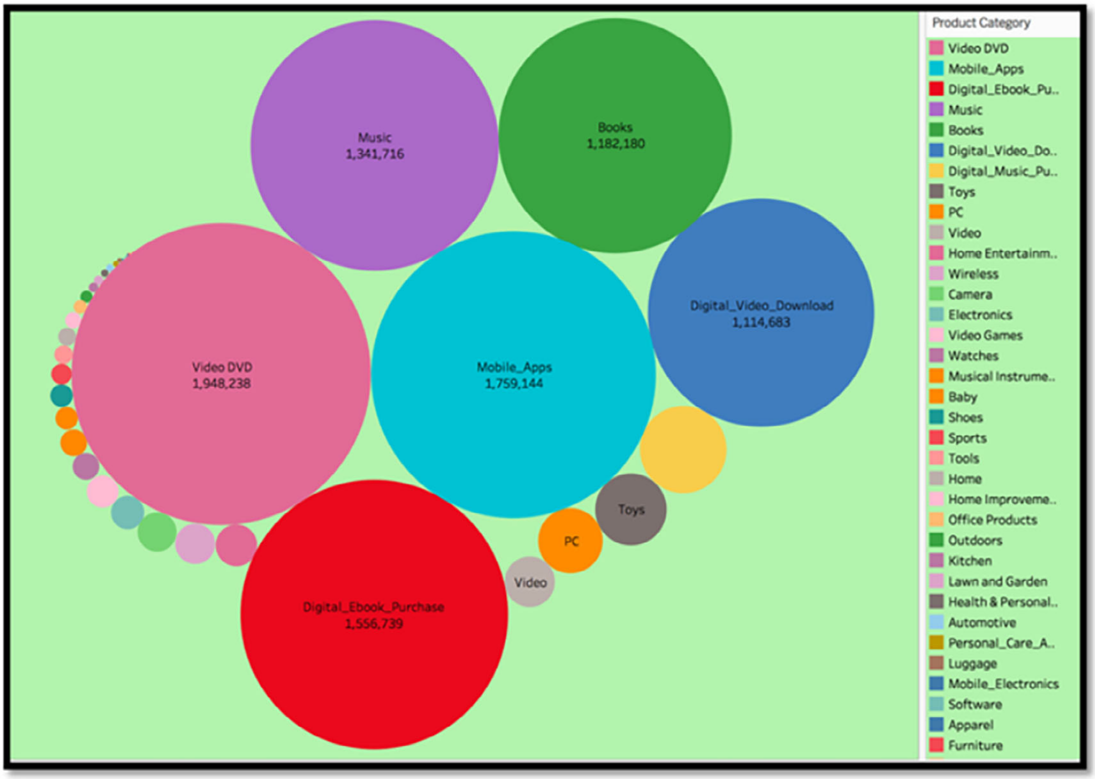

FIGURE 2 Review count by product category

with 1,556,739, and so on. It illustrates that overall digital products have received more reviews than the non-digital products.

5 | DATA ANALYSIS AND PREDICTION RESULTS

We implemented various models in Azure ML and Spark ML to predict the *star_rating*. As there is data size restriction in Azure ML and Databricks CE, we sampled the original dataset of 3.63 GB to 73 MB at a 2% rate of sampling on Azure ML by using Partition and Sample module. We used stratified sampling by selecting the *star_rating* column to ensure that the sampled dataset is a true representative of the original dataset. It took around 5.30 min for sampling.

The sampled dataset of 73 MB becomes a data source in both Azure ML and Spark ML on Databricks CE. We transfer the Spark ML codes of Databricks to Oracle BDCE for the full dataset. Since Oracle BDCE has enormous storage capacity in HDFS and Object Storage, we utilized the entire dataset of 3.63 GB to store and to compute using Spark ML.

5.1 | Matchbox recommender with Azure ML

The goal of the recommender is to provide Amazon customers with recommendations for product categories based on their previous ratings, as well as the ratings of other users. Moreover, the model has a feature to predict future ratings in the range of 0 to 5 by the user for a category. The dataset was split into training and testing fractions by 0.75 to 0.25 ratio. After the split, the training fraction was connected to Train Matchbox Recommender module and test fraction to four Score Recommender modules. Each of the four score recommenders represents different metrics: (a) item recommendation, (b) rating prediction, (c) similar items, and (d) similar users.

The model took 6 min to run. Table 2 shows the recommender models' evaluation result, which are root mean square error (RMSE), Mean Absolute Error, and normalized discounted cumulative gain.

TABLE 2 Recommenders evaluation

Recommendation type	Evaluation metrics and value	
Recommend items	NDCG 0.981748	
Predict ratings	MAE 0.714797	RMSE 1.312234
Related items	L1 Sim NDCG 0.930227	L2 Sim NDCG 0.934575
Related users	L1 Sim NDCG 0.800887	L2 Sim NDCG 0.785655

TABLE 3 Evaluation metrics for decision forest regression—Azure ML

	Cross validation model	Tune hyperparameters model
Negative log likelihood	64,491.782376	64,480.40652
Mean absolute error	0.885141	0.882887
Root mean squared error	1.145068	1.143119
Relative absolute error	0.991696	0.989171
Relative squared error	0.989869	0.986502
Coefficient of determination	0.010131	0.013498

TABLE 4 Evaluation metrics for boosted decision tree regression—Azure ML

	Cross validation model	Tune hyperparameters model
Mean absolute error	0.968952	0.649424
Root mean squared error	1.1154	0.911949
Relative absolute error	1.085597	0.727603
Relative squared error	0.93924	0.627851
Coefficient of determination	0.060706	0.372149

5.2 | Decision forest regression with Azure ML

We took a 2% sample of the original dataset and split the dataset into a 70:30 ratio for the training and testing. We used cross-validation and the tune model hyperparameters for the accuracy and generalization of the model when training. The permutation feature model is also used to find out the importance of various feature columns. Removing less important features resulted in the better performance of the decision forest regression model. The model takes 30 min to run, and it shows that the model with tune model hyperparameters provides a better result than the cross-validation model. Table 3 presents the result of the execution.

5.3 | Boosted decision tree regression with Azure ML

With the same 2% sample of the original dataset and a 70:30 ratio for the training and test, Table 4 shows that the tune model hyperparameters model provides a better result than the cross-validation model. The permutation feature model was also used to check the importance of various feature columns. Removing less important features did not improve the evaluation result of the boosted decision tree regression model. The model takes 1 hr and 30 min to run.

5.4 | Collaborative filtering recommender with Spark ML

Our Spark ML recommender on Databricks CE for data file size of 73 MB is based on collaborative filtering algorithm. The dataset used was a cleaned and transformed dataset created in Azure ML. The columns used are *customer_id*, *product_category*, and *star rating*. *Product_category* feature was transformed into integer using StringIndexer function. The target label is *star_rating*. The dataset of 73 MB is split to train and test fractions by 0.7 to 0.3 ratio. We have used alternating least squares algorithm to build the recommender.

Additionally, we have defined parameters and used fit() method to train the model. Then we tested the model to see the recommended category for each user. It takes around 30 min to build the model and evaluate it with the RMSE: 1.73.

We implemented the same code developed in Databricks CE to the Oracle BDCE for the full data size of 3.63 GB. It takes around 1 hr to train the model with an accuracy, RMSE: 2.10.

5.5 | Text analytics using logistic regression of Spark ML

For the text analysis, we used the classification model to predict if the sentiment will be positive or negative, which is the logistic regression algorithm in Spark ML. A rating higher than three is considered positive otherwise negative. We used the pipeline algorithm of Spark ML with Tokenizer to split the text into individual words, *StopWordsRemover* to remove common words such as “a” or “the” that have little predictive value. A *HashingTF* class is added to generate numeric vectors from the text values. The logistic regression algorithm is to train a binary classification model. So, the *stopwords* are removed from the tokens, and the model can predict the sentiments for the relevant text given by the sentiments dictionary of the words. The pipeline is used as an estimator and run with *fit()* method on training data to train the model. It takes around 4 min to run and gives an AUR of 0.71.

The code implemented in Oracle BDCE was the same code developed in Databricks CE but with the full dataset. The total data size of 3.63 GB was used to build the prediction model in the experiment. It takes around 20 min to train the model that gives an accuracy, AUR of 0.74.

5.6 | Decision tree regression with Spark ML

We used the decision tree regression model of Spark ML on the Databricks CE platform for the rating prediction. As most of the feature columns in the dataset are categorical, we used *StringIndexer* feature to index the features. *VectorAssemler* was also used to convert the features to numeric values. We split the dataset into a 70:30 ratio for training and testing. *CrossValidation* method is adopted for generalizing the model when training with the number of folds as 5. The pipeline is used as an estimator to vectorize the features and run with *fit()* method to train the model. Then, we evaluate the model using *RegressionEvaluator*. The RMSE of the model is 1.19, and it takes 5 min to develop and evaluate the model.

The same model was applied to Oracle BDCE for the full data, which results in the RMSE with 0.98, and it takes 1 hr and 20 min to develop and evaluate the model.

5.7 | Gradient boosted tree regression with Spark ML

For the rating prediction, we also adopt the gradient boosted tree regression algorithm with the same *StringIndexer* and *VectorAssemler* for the features and a 70:30 ratio for training and testing. In Databricks CE, it takes too long to train the model with *CrossValidation* method. So, we used *TrainValidation* Split method, which results in RMSE as 1.11, and it takes 15 min to build the model in Databricks CE. On Oracle BDCE, it takes 1 hr to develop the model with the same code for a data size of 3.63 GB and to show RMSE as 1.03.

6 | COMPARATIVE ANALYSIS

We implement various models in Azure ML and Spark ML to predict the *star_rating* in the range of 0 to 5. Tables 5–7 show the summary of the comparative analysis of all the ML models in Azure ML and Spark MLs on both Databricks CE and Oracle BDCE platform.

ML-platform	Model	RMSE	Time (min)
Azure ML (73 MB)	Matchbox recommender	1.22	6
Spark ML (73 MB)	Collaborative filtering	1.73	30
Spark ML (3.63 GB)	Collaborative filtering	2.10	60

TABLE 5 Comparison between different recommendation models

ML-platform	Model	RMSE	Time (min)
Azure ML (73 MB)	Decision forest regression	1.14	30
Azure ML (73 MB)	Boosted decision tree regression	0.91	90
Spark ML (73 MB)	Decision tree regression	1.19	5
Spark ML (73 MB)	Gradient boosted tree regression	1.11	15
Spark ML (3.63 GB)	Decision tree regression	0.98	80
Spark ML (3.63 GB)	Gradient boosted tree regression	1.03	60

TABLE 6 Comparison between different regression models

ML-platform	Model	AUR	Time (min)
Spark ML (73 MB)	Logistic regression	0.71	4
Spark ML (3.63 GB)	Logistic regression	0.74	20

TABLE 7 Comparison between different text analytics models

Table 5 shows that Azure ML has 28% better accuracy and the shorter computing time using Match Box Recommender with the sample dataset. However, with the full dataset, while Azure ML causes a memory error after a day, Spark ML in Oracle BDCE successfully generates a Collaborative Filtering model in 60 min. Even though Azure ML has 43% better RMSE than Spark ML in Oracle, Spark ML in Oracle works well with the entire dataset.

Table 6 presents that boosted decision tree regression model in Azure ML and gradient boosted decision the tree regression model has the better accuracy with the sample dataset but they need more computing time than the decision tree regression model. However, with the whole dataset, Spark ML in Oracle BDCE successfully generates a gradient boosted decision tree regression model in 60 min with 5% less accuracy than the decision tree regression model. However, Azure ML does not work with a memory error after several hours.

Table 7 illustrates that logistic regression model with the sample and the entire dataset has the almost same accuracy, that is, AUR. However, with the full dataset, Spark ML in Oracle BDCE successfully generates a logistic regression model in 20 min while in Databricks CE, it generates an error because the data is too big for a single server.

7 | CONCLUSION

We take and investigate the customer review dataset from Amazon and apply various ML models using Azure and Spark MLs to predict product ratings, make recommendations, and to predict customers' sentiments. Both Azure and Spark MLs are useful platforms for ML, especially for traditional ML with the small sample data and for Big Data massive dataset, respectively. It generates a memory error or takes more than a day to predict the ratings of the dataset that is greater than 500 MB when using traditional systems developed in Python and Azure ML. However, we show that you can predict the ratings in an hour if you use distributed parallel computing systems such as Hadoop and Spark, called Big Data.

A recommendation system is to help the user to discover products and content by predicting the user's rating of each item and showing them the items that they would rate highly. In our experiment, the recommender system from Azure ML provides an RMSE of 1.22 with the sample dataset. However, Spark ML can take the whole dataset as of 3.65 GB with a similar accuracy while Azure ML only computes its 73 MB sampled dataset.

Sentiment analysis allows companies to make sense of the unstructured text by automating business processes, getting actionable insights, and saving hours of manual data processing. The text analytics adopts the Logistic

Regression algorithm of Spark ML in the experiment on the Oracle BDCE platform, and it takes the entire dataset and computes a better result with an AUR of 0.74 than Spark ML on Databricks CE platform with the sample dataset.

We also have developed various regression models to predict the star rating in Azure ML and Spark ML. Rating prediction of products is vital for every business. It gives you an understanding of the product. Out of all the models, the best model was the boosted decision tree model in Azure ML, which gives a RMSE of 0.91 but it is only for the small sample dataset. The predictive models in Oracle BDCE cloud can be linearly scalable with much bigger datasets.

The paper illustrate that Big Data Predictive Analysis is possible with scalable computing systems. In this Big Data era, you can make business decisions to predict the trend and profit to perform shortly and accordingly using the scalable Big Data ML platform, which is not possible with the traditional systems, Azure ML.

ACKNOWLEDGMENTS

This study was supported by Oracle Cloud Innovation Accelerator. Oracle Big Data Cloud Service was used in the data analysis.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

AUTHOR CONTRIBUTIONS

Monika Mishra: Conceptualization; data curation; formal analysis; investigation; methodology; software; writing-original draft. **Jongwook Woo:** Conceptualization; funding acquisition; investigation; methodology; project administration; resources; software; writing-review and editing.

ORCID

Jongwook Woo https://orcid.org/0000-0003-3524-8906
Monika Mishra https://orcid.org/0000-0003-2696-3262

RELATED WIRES ARTICLE

A survey on graphic processing unit computing for large-scale data mining

FURTHER READING

He, R., & McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web* (pp. 507–517). International World Wide Web Conferences Steering Committee.

McAuley, J., Targett, C., Shi, Q., & Van D. H. A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 43–52). ACM.

REFERENCES

García-Gil, D., Ramírez-Gallego, S., & García, S. (2017). A comparison on scalability for batch big data processing on Apache Spark and Apache Flink. *Big Data Analytics*, *2*, 1. https://doi.org/10.1186/s41044-016-0020-2

Gupta, N., Le, H., Boldina, M., & Woo, J. (2019). Predicting fraud of AD click using Traditional and Spark ML. In *KSII The 14th Asia Pacific International Conference on Information Science and Technology (APIC-IST)* (pp. 24–28).

Patel, B. (2017). Predicting Amazon product reviews' ratings. *Towards Data Science* (April 27). https://towardsdatascience.com/predicting-sentiment-of-amazon-product-reviews-6370f466fa73

Pritwani, K., Wasley, K., & Woo, J. (2018). Spark Big Data analysis of world development indicators. *Global Journal of Computer Science and Technology*, *18*, 1–10.

Purushu, P., Melcher, N., Bhagwat, B., & Woo, J. (2018). Predictive Analysis of Financial Fraud Detection using Azure and Spark ML. *Asia Pacific Journal of Information Systems (APJIS)*, *28*(4), 308–319.

Qasem, M., Thulasiram, R., & Thulasiram, P. (2015). Twitter sentiment classification using machine learning techniques for stock markets. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 834–840). Institute of Electrical and Electronics Engineers.

S3.amazonaws.com. (n.d.). *Amazon reviews multilingual dataset*. Available from https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_US_v1_00.tsv.gz

Saumya, S., Singh, J. P., Baabdullah, A. M., Rana, N. P., & Dwivedi, Y. K. (2019). *Ranking online consumer reviews*. arXiv:1901.06274.

Woo, J., & Xu, Y. (2011). Market basket analysis algorithm with map/reduce of cloud computing. In *International conference on parallel and distributed processing techniques and applications (PDPTA 2011), Las Vegas*.

Woolf, M. (2017). Playing with 80 million Amazon product review ratings using Apache Spark. *Minimaxir* (January). https://minimaxir.com/2017/01/amazon-spark/

How to cite this article: Woo J, Mishra M. Predicting the ratings of Amazon products using Big Data. *WIREs Data Mining Knowl Discov*. 2020;e1400. https://doi.org/10.1002/widm.1400