Scalable Predictive Analysis for Airbnb Listing Rating

Savita Yadav, Samyuktha Muralidharan, +Sanghoon Lee, Jongwook Woo
Department of Information Systems, California State University
Los Angeles, California
+Department of Electrical Engineering, Yonsei University
Seoul, Korea
[e-mail: {syadav5, smurali2, jwoo5}@calstatela.edu, slee@yonsei.ac.kr]
*Corresponding author: Jongwook Woo

Abstract

This paper presents predictive models for Airbnb Rating using the traditional and Big Data platforms such as Azure ML Studio and Spark ML. We aim to use Machine Learning algorithms in each platform to build models to predict a rating of the Airbnb listing. The Airbnb ratings are the best constructive feedback that is helpful for hosts to alter their business strategy to target more potential customers. We use Two-Class Classification models to predict if the listing has a high or low rating based on the features of the listing. We compare the evaluation metrics of the traditional and Big Data approaches for rating prediction and discuss the performance of the models. We built Big Data models using Spark Cluster, which follows the process of distributed parallel computing.

Keywords: Airbnb Rating, Azure ML, Spark ML, Classification, Big Data, Predictive Analysis, neural network.

1. Introduction

Airbnb is an online marketplace that connects people who want to rent out their homes with people looking for accommodations in that locale.

The objective of the project deals with predicting the rating of the Airbnb Listing. We aim to build a model that predicts if a property has a high or a low rating based on the features of the listing. It helps the hosts to know if their property is suitable and how their listing compares to other similar listings. It helps them make simple changes to the properties they are listing to boost customer satisfaction. It can be an opportunity for the hosts to improve the listing and the hospitality they provide.

It can serve as a baseline to understand the factors that contribute to the popularity and rating of a listing. We make use of Airbnb Listings data, which consists of various Airbnb properties and their features. We develop a Big Data cluster to store the data set and process data engineering to filter the data set to train and build prediction models properly.

2. Related Work

Athor et al. predict Airbnb customers' behavior in classifying their reviews as high or low-rated using associated attributes and topics from the reviews. They used the Tree Model, Random Forest Model, Least Absolute Shrinkage and Selection Operation and Logistic Regression

This research was supported by a research credit from Databricks University Alliance and by NRF of Korea

Model, Artificial Neural Network, and Multi-Layer Perceptron to classify prediction in a traditional system with small data set - 66,630 reviews [1].
We developed prediction models for Airbnb Rating based on the Big Data Spark cluster with a large-scale data set.

3. Background Work

We can use Classification Machine Learning algorithms to build predictive models for rating prediction. We used binary classification models for rating prediction. Our models are to classify the listings as either high/low rated. We can quickly build legacy machine learning models with a small data set as a cloud computing service using Azure ML Studio. Azure ML supports many classification models.
We can improve the accuracy and generalization of the models with Tune Model Hyperparameters, Cross-Validation, and Permutation feature importance modules. The evaluation metrics used are Area Under Curve (AUC) and Precision. Spark ML is a Big Data solution for predictive analysis supporting classification libraries. We also use Binary Classification Evaluator to evaluate the models.

4. Experimental result

We implemented various machine learning models to predict rating in Azure ML and Spark ML. **Review Scores Rating** column of the data indicates the overall rating of a listing. Initial processing of the original dataset involves converting the **Review Scores Rating** column to a categorical column. Thus, we redefine and classify the listings as high-rated with a value, 1 when '**Review Scores Rating**' >= 80. And others are as low-rated with a value, 0.
As a Big Data engineering process in the Hadoop cluster, we filter the dataset using Hive QL to contain only the Airbnb listings in the United States [2]. Thus, we can build the sample and entire data sets as 30 MB and 400 MB respectively from 4GB, and then train and implement models for rating prediction. To perform predictive analysis with the legacy machine learning models of Microsoft Azure ML Studio, we used the sample dataset, 30 MB. We used the Spark Computing engine of the Hadoop-Spark cluster with 400 MB in PySpark and Spark ML library to build predictive models.

4.1 Legacy Predictive Analysis with Azure ML

The sample 30 MB dataset is split into training and testing set by 50:50 ratio. The Machine Learning models are implemented with the three algorithms: Two-Class Logistic Regression, Two-Class Decision Forest, and Two-Class Boosted Decision Tree.
Table 2 shows the evaluation metrics of the models. The evaluation result of the model trained with Two-Class Logistic Regression shows an AUC of 0.996, Precision of 0.988, and it took 6 minutes to run this experiment. The result of the Two-Class Decision Forest is observed with an AUC of 0.977, Precision of 0.992, and it took 4 minutes to run this experiment. Compared to the other two models, Two-Class Boosted Decision Tree shows better results in AUC. Thus, we got an AUC of 0.998, Precision of 0.978, but the computing time takes longer with 45 minutes to run.

Table 1. Evaluation metrics for Rating Prediction in AzureML

Metrics	Two-Class Logistic Regression	Two-Class Decision Forest	Two-Class Boosted Decision Tree
Accuracy	0.983	0.919	**0.986**
Recall	0.954	0.725	**0.972**
Precision	0.988	**0.992**	0.978
AUC	0.996	0.977	**0.998**
Training Time	6 mins	**4** mins	45 mins

The important metrics to evaluate models are AUC and Precision. We notice that the Two-Class Decision Forest Classifier has a higher precision value of 0.992 than the other models. But, Two-Class Boosted Decision Tree has the highest AUC of 0.998 even though the training time takes 45 minutes.

4.2 Big Data Predictive Analysis with Spark ML

We develop models with Spark ML, composed in a Big Data cluster of Databricks Enterprise Edition (EE). The algorithms used to build

two-class classification models for rating prediction are Decision Tree, Random Forest (RF), Gradient Boosted Tree (GBT), Logistic Regression, Multilayer Perceptron (MLP), Factorization Machine (FM), and Support Vector Machine (SVM).

The initial steps involve importing the dataset and reading the file as a PySpark data frame. We convert the Rating column, i.e., the label in the dataset, to a categorical column with two values 0 (low-rated) and 1 (high-rated). We preprocess the data to remove outliers and handle null values. We split the dataset into a 70:30 ratio for training and testing data. We defined a pipeline for feature transformation and training the classifier model. The pipeline consists of a String Indexer, Vector Indexer, MinMax Scaler, Vector Assembler, and a two-class Classifier algorithm that trains a Binary Classification model.

The essential metrics to evaluate the models are AUC and Precision, as we do in Azure ML. AUC is the measure of the ability of a classifier to distinguish between classes. The higher the AUC, the better the performance of the model. Precision indicates how many positive predictions were correct.

Table 2. Models for Rating Prediction

Models	Computing Time	Precision	Recall	AUC
Decision Tree	**1.15 mins**	**0.983**	0.984	0.972
Random Forest	2.13 mins	**0.985**	0.993	**0.979**
GBT	2.35 mins	**0.984**	0.993	**0.977**
Logistic Regression	1.47 mins	0.968	**0.998**	0.959
Multilayer Perceptron	3.21 mins	**0.983**	0.991	**0.975**
Factorization Machines	1.70 mins	0.972	**0.997**	0.965
SVM	4.49 mins	0.966	**0.998**	0.958

Table 2 shows that the AUC of all models is closer to 1. RF, GBT, and MLP classifiers have almost similar AUC values (0.975 – 0.979) compared to others. For Precision, we observe that Decision Tree, RF, GBT, and MLP classifiers have slightly higher values (0.983 – 0.985). The computing time in the Decision Tree is the most efficient, with 1.15 mins.

We aim to reduce the number of False Positives, so we should consider the models that give a higher Precision value. As we need to help, the Airbnb hosts understand if their property is suitable. Although the computing time to run the MLP Classifier and SVM is slightly higher than traditional machine learning models, the results seem to be good. We can conclude that RF, GBT, and MLP models have performed well for classified listings in the listings of the states. Furthermore, RF and GBT have shorter computing times than MLP.

5. Conclusions

The paper attempts to come up with the best-performing models for predicting Airbnb Rating. We performed the legacy and Big Data predictive analyses using AzureML Studio and Spark ML in Databricks EE platforms, respectively. We used three two-class classifier algorithms in Azure ML and seven algorithms in Spark ML for rating prediction. We showed that the Big Data Spark ML has a much faster computing time than the legacy approach, even with more than ten times more enormous data set.

Decision Forest and Boosted Tree classifiers with the sample 30 MB data in Azure ML are optimal models to classify listings. But, with the entire US data set, RF and GBT models generate the optimal Precision and AUC with less computing time than MLP. Thus, these models can be an excellent fit for predicting the Airbnb listings ratings with the best accuracy and proper training time.

References

[1] Athor, S., & Marcel, C. (2021). Rating prediction of peer-to-peer accommodation through attributes and topics from customer review. Journal of Big Data, 8(1).

[2] Airbnb Listings Dataset (2021). Retrieved from https://public.opendatasoft.com/explore/dataset/airbnb-listings/table/?disjunctive.host_verifications&disjunctive.amenities&disjunctive.features