# Predicting Airbnb Listing Price and Rating using AzureML and SparkML

Samyuktha Muralidharan, Sanjana Boddireddy, Savita Yadav, Farnood Rahbar Far
Department of Information Systems, California State University
Los Angeles
Tel. 424-345-5537, Fax. 323-343-5209
e-mail: smurali2@calstatela.edu; sboddir@calstatela.edu; syadav5@calstatela.edu; frahbar@calstatela.edu

**Abstract:** This paper aims to build predictive models for Airbnb Price and Rating using the platforms AzureML Studio, Databricks and Hadoop-Spark cluster. We made use of Airbnb Listings and Reviews datasets for predictive analysis. Our goal is to use three Machine Learning algorithms in each platform to build models to predict the price of the Airbnb Listing as well as if the listing has a high/low rating. We used Regression models for predicting the Listing price and Two-Class Classification models to predict if the listing has a high/low rating. We make a comparison of the evaluation metrics of the models of each prediction and determine which model performed better. We also discuss the efficiency of performing predictive analysis in Hadoop-Spark Cluster, that follows the approach of distributed parallel computing.

**Keywords:** *Airbnb Price, Airbnb Rating, AzureML, Databricks, SparkML, Hadoop, Regression, Classification.*

## 1.Introduction

Airbnb is an online marketplace that connects people who want to rent out their homes with people who are looking for accommodations in that locale. It helps people to list, explore and book unique properties all over the world. In this project we are predicting Airbnb Listing Price and Rating of the Listing. The objective of part one of the project includes building a model that predicts the optimal price of a property considering the features of the listings. One challenge that most of the Airbnb hosts face is determining the optimal rent price per night. Usually, when users look for properties, they are presented with a good selection of listings. The amount a host can charge on a nightly basis is closely linked to the dynamics of the marketplace. If the host charges above the market price, then users will select other affordable alternatives. If the nightly rent price is set too low, the hosts will miss out on potential revenue.So, we use machine learning models to predict the optimal prices that the hosts can set for their properties. It helps hosts to better understand how different features of the listing can be used to accurately predict the price. With better price suggestion estimates, Airbnb home providers can reach an equilibrium price that optimizes profit and affordability.

The second part 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 good and how their listing compares to other similar listings. It helps them to make simple changes to the properties they are listing in order to boost customer satisfaction. It can be an opportunity for the hosts to make improvements to 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. Our dataset is of the size 4GB and in csv format. We make use of Airbnb Listings data, that consists of various Airbnb properties and its features. The second dataset used is the Airbnb Reviews, that has the user reviews for each listing.

## 2. Technical Specifications

We have implemented this project on three platforms. The first is Microsoft Azure Machine Learning Studio. Here we used the sampled dataset, of size 30MB, to perform predictive analysis. The second platform is the Databricks-community edition. Here we used the Spark Computing engine, PySpark and SparkML library to build predictive models for the sampled dataset. We also used the Hadoop-Spark cluster at SCU in Korea to build predictive models on the whole dataset, using Spark Computing engine that is integrated with the Hadoop cluster, and PySpark and SparkML library. Table 1 shows the technical specifications of all the three platforms.

Table 1. Technical Specifications

| Azure ML Studio | Databricks | Hadoop Cluster |
|---|---|---|
| **Workspace Type:** Free<br>**Storage:** 10GB<br>**Nodes:** 1<br>**Region:** South Central US | **Databricks Runtime version:** DBR 8.1 (includes Apache Spark 3.1.1, Scala 2.12)<br>**Memory:** 15.3 GB<br>2 cores, 1 DBU<br>**Nodes:** 1<br>**File System:** DBFS (Data Bricks File System)<br>**Python Version** 3.8.6 | **Cluster version:** Hadoop 3.1.1.3.1.4.0-315<br>**Nodes:** 3<br>**HDFS Capacity:** 11.7 TB<br>**RAID:** 24TB/18TB<br>**Python version:** 2.7.5<br>**Spark version:** 2.3.2.3.1.4.0-315<br>**Apache Hive version:** 3.1.0.3.1.4.0-315<br>**CPU cores**: 48<br>**Clock:** 842.230 MHz<br>**Memory:** 192GB |

## 3. Related Work

We have built prediction models for Airbnb Price and Rating based on the following papers,

[1] A paper by Kalehbasti et al. (2019) attempts to come up with the best-performing model for predicting the Airbnb prices for NYC, based on a limited set of features. ML techniques like linear regression, SVR, and neural networks along with feature importance analyses are used to achieve

the best results. They also performed sentiment analysis of the reviews using TextBlob sentiment analysis library, while we made use of Apache Hive to perform sentiment analysis.

[2] Choudhary et al. (2018) analyzes Airbnb listings in the city of San Francisco to better understand how different attributes of the listings can be used to accurately predict the price. They used the Random Forest Regressor to predict the price of the listings. They also focused on the availability of listings by segregating them into high/low availability using Clustering and Naïve Bayes algorithm.

[3] A study by Luo (2018), aims to identify the weights of latent lodging experience aspects that customers consider to form their overall ratings based on the 8 basic emotions. This study applied aspect-based sentiment analysis and the latent aspect rating analysis (LARA) model to predict the aspect ratings and find the latent aspect weights. The NRC Emotion Lexicon tool with 8 emotions was used to assess the sentiments associated with each lodging aspect. By applying latent rating regression the predicted overall ratings were calculated. In our project we used Apache Hive to assess the sentiment of the user reviews and classified the listings as either high/low rated using two-class classifier algorithms.

The above papers have implemented prediction models using Python and in platforms such as Jupyter Notebook. Whereas our project deals with building ML Models using AzureML and Databricks/Hadoop-Spark cluster by utilizing Spark computing engine, PySpark and SparkML library.

## 4.Background/ Existing Work

In our project we have used three Machine Learning algorithms in each platform to build predictive models for price and rating prediction.

### 4.1 Hive Sentiment Analysis

There are user reviews available for the listings in the Airbnb Reviews dataset. In order to utilize the user reviews in building predictive models, we determined the sentiment of the reviews by performing sentiment analysis using Apache Hive, which belongs to the Hadoop ecosystem. Hive gives a SQL-like interface to query data and the language used is HiveQL, which is similar to SQL. We determined the overall sentiment of each listing Id so that we can utilize the sentiment of the listings as a feature in building our models.

### 4.2 Regression

Regression is a supervised machine learning technique which is used to predict continuous values. In part one of our project involving price prediction, we make use of Regression models, as price, the target variable is a continuous numeric variable. We used three algorithms in AzureML- Decision Forest Regression, Boosted Decision Tree and Bayesian Linear Regression. The process is similar to the lab work involving ‘energy efficiency’ dataset. We used Tune Model Hyperparameters module for improved model performance, Cross Validate module to see if the model generalizes well and Permutation feature importance module to eliminate less important features iteratively.

For price prediction using SparkML we used the algorithms - Gradient Boosted Tree Regression, Decision Tree Regression, Random Forest Regression. The regression models used are based on the lab work involving predicting the ‘arrival delay’ in the ‘flights’ data. The similar process of creating a pipeline for feature transformation and training a regression model was performed. We also used a Cross Validator to find the best performing parameters. The evaluation metrics used are Root Mean Square Error (RMSE) and Co-efficient of Determination(R2).

### 4.3 Classification

Classification is a supervised ML technique which categorizes a set of data into classes. We used binary classification models in part two of our project involving rating prediction. We build a model to classify the listings as either high/low rated. In AzureML we used three classification models - Two-class Logistic Regression, Two-class Decision Forest, Two-class Boosted Decision Tree. The process is similar to the lab work involving categorizing diabetes patients. We used Tune Model Hyperparameters, Cross Validate and Permutation feature importance modules. The evaluation metrics used are AUC (Area Under Curve) and Precision.

The algorithms used in SparkML - Logistic Regression, Decision Tree Classifier, Gradient Boosted Tree Classifier. The process is similar to the lab work involving predicting if a flight arrives late or not. Here we classified the flights with delay time more than 15 minutes as ‘delayed’ and others as ‘not delayed’. We followed a similar approach in our project by classifying the listings with ‘Review Scores Rating’>=80 as high-rated (1) and others as low-rated (0). We used Binary Classification Evaluator to evaluate the model.

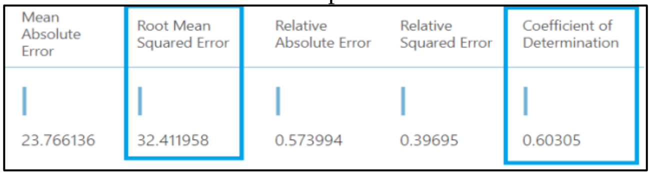
## 5. Our Work

We implemented various machine learning models in AzureML and using SparkML to predict price and rating. We made use of a sampled dataset of size 30MB to perform predictive analysis in AzureML Studio and Databricks. For the SCU Hadoop Cluster, we made use of the whole dataset to build predictive models.

### 5.1. Price Prediction

This involves building machine learning models to predict the price per night of Airbnb Listings.

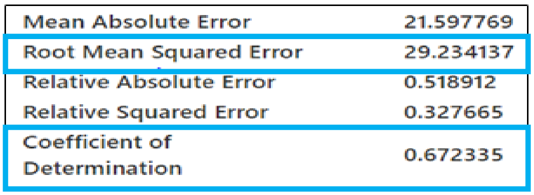
**Decision Forest Regression (AzureML):** We used a sampled file of size 30 MB and split the data as 70% train and 30% test. Tune Model Hyperparameters was used to find the best performing model. Permutation feature importance module was used to determine the best features to use in the model and Cross validate module to generalize the model. Figure 1 shows the evaluation results for Decision Forest Regression. We got RMSE of 32.4, R2 of 0.6 and it took 1.5 minutes to run this experiment.



| Mean Absolute Error | Root Mean Squared Error | Relative Absolute Error | Relative Squared Error | Coefficient of Determination |
|---|---|---|---|---|
| 23.766136 | 32.411958 | 0.573994 | 0.39695 | 0.60305 |

Figure 1. Evaluation results for Decision Forest

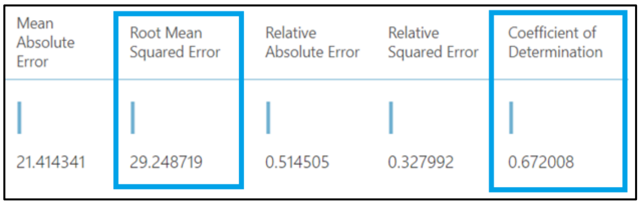
**Boosted Decision Tree Regression (AzureML):** Figure 2 shows the evaluation results. We got an RMSE of 29.2, R2 of 0.67 and it took 2 minutes to run this experiment.



| | |
|---|---|
| Mean Absolute Error | 21.597769 |
| Root Mean Squared Error | 29.234137 |
| Relative Absolute Error | 0.518912 |
| Relative Squared Error | 0.327665 |
| Coefficient of Determination | 0.672335 |

Figure 2. Evaluation results for Boosted Decision Tree

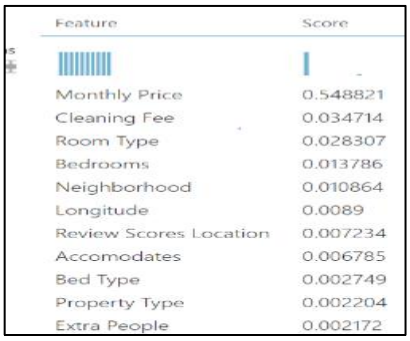
**Bayesian Linear Regression (AzueML):** Figure 3 shows the evaluation results. We got an RMSE of 29.2, R2 of 0.67 and it took 2 minutes to run this experiment.



| Mean Absolute Error | Root Mean Squared Error | Relative Absolute Error | Relative Squared Error | Coefficient of Determination |
|---|---|---|---|---|
| 21.414341 | 29.248719 | 0.514505 | 0.327992 | 0.672008 |

Figure 3. Evaluation results for Bayesian Linear Regression

Figure 4 shows the final results of the Permutation Feature Importance Module of Bayesian Linear Regression. These are the features that affect the Price of the Airbnb Listing and thus are important for Price Prediction. From Table 2, we see that the best results with less RMSE (29.2) and better R2(0.67) values is provided by Bayesian Linear and Boosted Decision Tree Regression models.



| Feature | Score |
|---|---|
| Monthly Price | 0.548821 |
| Cleaning Fee | 0.034714 |
| Room Type | 0.028307 |
| Bedrooms | 0.013786 |
| Neighborhood | 0.010864 |
| Longitude | 0.0089 |
| Review Scores Location | 0.007234 |
| Accomodates | 0.006785 |
| Bed Type | 0.002749 |
| Property Type | 0.002204 |
| Extra People | 0.002172 |

Figure 4. Important Features for Price Prediction

Table 2. Comparison of Metrics for Azure Price Prediction

| | Decision Forest Regression | Boosted Decision Tree Regression | Bayesian Linear Regression |
|---|---|---|---|
| **RMSE** | 32.411958 | 29.234137 | 29.248719 |
| **R2** | 0.60305 | 0.672335 | 0.672008 |

**SparkML – Databricks:** The algorithms used to build regression models for price prediction are Gradient Boosted Tree, Decision Tree and Random Forest Regression models. The initial steps involve importing the dataset and reading the file as a pyspark data frame. We preprocess the data to remove outliers and handle null values. We split the dataset into 70:30 ratio for training and testing data.

We defined a pipeline for feature transformation and training the regression model. The pipeline consists of a String Indexer, Vector Indexer, MinMax Scaler, Vector Assembler and a Regression algorithm that trains a Regression model. We used the Cross Validator class to find the best performing parameters. For all the three algorithms of price prediction the number of folds for cross validator is assigned to 5. The model produced by the pipeline was applied on the test data to generate predictions.

Table 3. Evaluation metrics for Databricks Price Prediction

| Databricks (1 node) | Decision Tree Regression | Gradient Boosted Tree Regression | Random Forest Regression |
|---|---|---|---|
| **RMSE** | 39.30215 | 36.34804 | 38.02226 |
| **R2** | 0.652791 | 0.657432 | 0.680739 |
| Time taken | 10 minutes | 40 minutes | 60 minutes |

Root Mean Square Error (RMSE) gives the average difference between the predicted and actual values. It is in the same unit as the target variable. So, here RMSE indicates the average difference in dollars between predicted and actual price of the listings. Coefficient of Determination (R2) indicates how well the model fits the data. It evaluates the scatter of the data points around the fitted regression line.

Table 3 shows that Random Forest Regression has given a higher R2 value (0.68) and the RMSE (38) is almost similar to the other models. Even though the time taken to build and evaluate the model is longer for Random Forest, the results seem to be good. Gradient Boosted Tree Regression is with the least RMSE (36), and R2 value of 0.657 is slightly lesser than Random Forest Regression model. Also, the time taken to run is lesser than Random Forest Regression. So, we can conclude that Random Forest and GBT Regression models give good results for Price Prediction in Databricks.

**SparkML – SCU Hadoop Cluster:** We also use the spark computing engine that is integrated with the Hadoop Cluster to build predictive models for price. We used the entire dataset that is stored in HDFS and the same ML algorithms that we used for price prediction in Databricks.

Table 4. Evaluation metrics for Price Prediction in Hadoop

| Hadoop Cluster (3 nodes) | Decision Tree Regression | GBT Regression | Random Forest Regression |
|---|---|---|---|
| **RMSE** | 37.550365 | 37.760964 | 36.154873 |
| **R2** | 0.6618076 | 0.6726220 | 0.691347 |
| Time taken | 5 minutes | 25 minutes | 10 minutes |

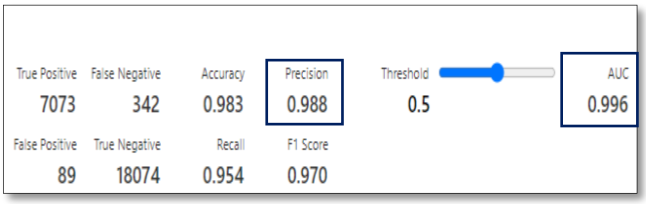
Table 4 shows that the evaluation metrics for the three models in Hadoop Cluster is almost similar to the results in Databricks. Random Forest Regression has performed the best with the least RMSE (36) and the highest value of R2 (0.69). The time taken to build and evaluate this model was 10 minutes.

When comparing tables 3 and 4, we see that there is a great difference in the time taken to build the models. This is because Databricks has only one node and all the computation is carried out in that node. Hadoop Cluster consists of three nodes. So, data is distributed across all the three nodes and processed parallelly. Thus, we could see a significant decrease in the time taken to build and evaluate predictive models in Hadoop-Spark cluster.

## 5.2 Rating Prediction

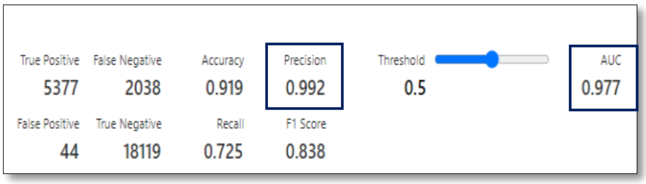
In part two of our project, we aim to build a model to classify the listings as high-rated or low-rated. Initial processing of the original dataset involves converting the Review Scores Rating column (this column indicates the overall rating of a listing) to a categorical column i.e., classifying the listings with 'Review Scores Rating' >= 80 as high-rated (1) and others as low-rated (0).

**Two-Class Logistic Regression (AzureML):** The dataset was split into training and testing set by 50:50 ratio.We used the Tune Model Hyperparameters module for training and the Cross Validate model to see if the model generalizes well. We used permutation feature importance to determine the best features to use in a model based on scores. Figure 5 shows that we got an AUC of 0.996, Precision of 0.988 and it took 6 minutes to run this experiment.



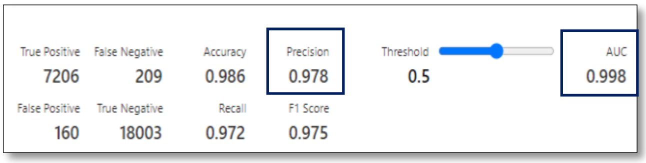
| | | | | | |
|---|---|---|---|---|---|
| True Positive | False Negative | Accuracy | Precision | Threshold | AUC |
| 7073 | 342 | 0.983 | 0.988 | 0.5 | 0.996 |
| False Positive | True Negative | Recall | F1 Score | | |
| 89 | 18074 | 0.954 | 0.970 | | |

Figure 5. Evaluation results for Logistic Regression



| | | | | | |
|---|---|---|---|---|---|
| True Positive | False Negative | Accuracy | Precision | Threshold | AUC |
| 5377 | 2038 | 0.919 | 0.992 | 0.5 | 0.977 |
| False Positive | True Negative | Recall | F1 Score | | |
| 44 | 18119 | 0.725 | 0.838 | | |

Figure 6. Evaluation results for Two-Class Decision Forest



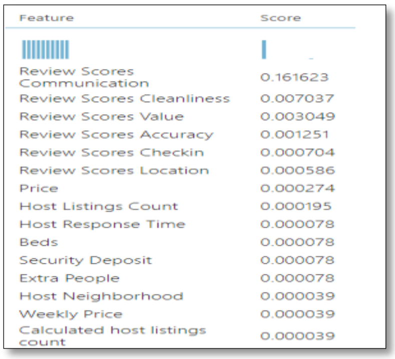
| | | | | | |
|---|---|---|---|---|---|
| True Positive | False Negative | Accuracy | Precision | Threshold | AUC |
| 7206 | 209 | 0.986 | 0.978 | 0.5 | 0.998 |
| False Positive | True Negative | Recall | F1 Score | | |
| 160 | 18003 | 0.972 | 0.975 | | |

Figure 7. Results for Two-Class Boosted Decision Tree

**Two-Class Decision Forest (AzureML):** Figure 6 shows that we got an AUC of 0.977, Precision of 0.992 and it took 4 minutes to run this experiment.

**Two-Class Boosted Decision Tree (AzureML):** Compared to the other two models, Two-Class Boosted Decision Tree gave better results after removing less important features. Figure 7 shows that we got an AUC of 0.998, Precision of 0.978 and it took 45 minutes to run.

The important metrics to evaluate are AUC and Precision (as we should reduce the number of False Positives). We notice that the Two-Class Decision Forest Classifier has a higher precision value of 0.992 than the other models.



| Feature | Score |
|---|---|
| Review Scores Communication | 0.161623 |
| Review Scores Cleanliness | 0.007037 |
| Review Scores Value | 0.003049 |
| Review Scores Accuracy | 0.001251 |
| Review Scores Checkin | 0.000704 |
| Review Scores Location | 0.000586 |
| Price | 0.000274 |
| Host Listings Count | 0.000195 |
| Host Response Time | 0.000078 |
| Beds | 0.000078 |
| Security Deposit | 0.000078 |
| Extra People | 0.000078 |
| Host Neighborhood | 0.000039 |
| Weekly Price | 0.000039 |
| Calculated host listings count | 0.000039 |

Figure 8.Important Features for Rating Prediction

**SparkML – Databricks:** The algorithms used to build two-class classification models for rating prediction are Logistic Regression, Decision Tree Classifier and Gradient Boosted Tree Classifier. 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 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 'number of folds' of the Cross Validator is set to 5 for Logistic Regression and Decision Tree Classifier, and for GBT Classifier it is set to 2.

The important metrics to evaluate the models are AUC (Area Under Curve) and Precision. AUC is the measure of the ability of a classifier to distinguish between classes. The higher the AUC, better the performance of the model.

Table 5. Metrics for Rating Prediction in Databricks

| Databricks (1 node) | Logistic Regression | Decision Tree Classifier | GBT Classifier |
|---|---|---|---|
| **AUC** | 0.9553161 | 0.9585590 | 0.9585721 |
| **Precision** | 0.966690 | 0.970989 | 0.970828 |
| Time taken | 15 minutes | 25 minutes | 30 minutes |

Precision indicates how many positive predictions were actually correct. In this case, we aim to reduce the number of False Positives, as we need to help the Airbnb hosts understand if their property is good. A False Positive i.e., a false prediction that the new listing's rating is high, would make the hosts believe that no changes are required for the property and there are chances that they would not make an effort to improve the features of the property. This could result in decreased customer satisfaction and negative user reviews for the property, affecting the hosts bookings.

Table 5 shows that the AUC of all the three models is closer to 1 and they are almost similar in value (0.95). We see that the Gradient Boosted Tree and Decision Tree Classifiers have slightly higher precision values (0.97) than Logistic Regression. As our aim is to reduce the number of False Positives, we should consider the models that give a higher Precision value. Although the time taken to run the GBT and Decision Tree Classifiers is slightly higher than Logistic Regression, the results seem to be good. We can conclude that both GBT and Decision Tree Classifier models have performed well for classifying listings in Databricks.

**SparkML – SCU Hadoop Cluster:** We also make use of the spark computing engine that is integrated with the Hadoop Cluster to build predictive models for rating. We use the entire dataset that is stored in HDFS to build models.

Table 6 shows that the evaluation metrics for the three models in Hadoop - Spark Cluster is almost similar to the results in Databricks. The AUC of all the three models is close to one (0.95) and the GBT Classifier and Decision Tree Classifier models have slightly higher Precision values (0.97) than Logistic Regression. The number of folds used for the Cross Validator for all the three algorithms is 5. Gradient Boosted Tree Classifier took around 60 minutes to run. This time is much greater than the other two models. So, in terms of AUC, Precision and Time taken Decision Tree Classifier has performed well.

When comparing tables 5 and 6, the models Logistic Regression and Decision Tree Classifier have taken very less time to build and evaluate model in the Hadoop Cluster. The time taken by Gradient Boosted Tree Classifier is much higher for the whole dataset as we made use of higher number of folds (K=5) for the Cross Validator as compared to just 2 folds for the sampled data in Databricks.

Table 6. Evaluation metrics for Rating Prediction in Hadoop

| Hadoop Cluster (3 nodes) | Logistic Regression | Decision Tree Classifier | GBT Classifier |
|---|---|---|---|
| AUC | 0.9504987 | 0.9590108 | 0.955631 |
| Precision | 0.9638829 | 0.9702456 | 0.9692927 |
| Time taken | 6 minutes | 7 minutes | 60 minutes |

The time to build and evaluate a model for the sampled data in Databricks is much higher when compared to the time taken for the whole dataset in Hadoop. This is because Databricks has only one node and all the computation is carried out in that node. Hadoop Cluster consists of three nodes. So, data is distributed across all the three nodes and processed parallelly. Thus, we could see a significant decrease in the time taken to build and evaluate predictive models in Hadoop-Spark cluster. So Big Data platforms like Hadoop can be utilized for predictive analysis of big data with very less computation time.

## 6. Conclusion

This paper attempts to come up with the best performing models for predicting Airbnb Listing price and Rating. We performed predictive analysis in three platforms-AzureML Studio, Databricks and Hadoop-Spark Cluster. We used three Regression algorithms for price prediction and three two-class classifier algorithms for rating prediction in each of the three platforms. Tables 7 and 8 show the best performing models among the models tested, for Price and Rating Prediction in AzureML and SparkML.

Table 7. Best performing models for Price Prediction

| | Algorithm | RMSE | R2 |
|---|---|---|---|
| **AzureML** | Bayesian Linear Regression | 29.248719 | 0.672008 |
| **SparkML** | Random Forest Regression | 38.02226 | 0.680739 |

For Price Prediction (Table 7), we have achieved an RMSE of 29 for Bayesian Linear Regression in AzureML and 38 in SparkML (RFR). The R2 value achieved for price prediction is 0.68 that is quite high for price related ML models. The best models to classify listings as high-rated or low-rated (Table 8) have given an AUC of 0.996 for Logistic Regression in AzureML and 0.95 in SparkML(GBT). The Precision value achieved is quite high with around 0.98. This level of accuracy is a promising outcome and these are the best performing models that have given phenomenal results.

Table 8. Best performing models for Rating Prediction

| | Two-class Classifier | AUC | Precision |
|---|---|---|---|
| **AzureML** | Logistic Regression | 0.996 | 0.988 |
| **SparkML** | GBT | 0.9585 | 0.970828 |

Predicting Airbnb Listing Price helps hosts to set the optimal price for their properties and it also helps them to understand how different features of the listing can be used to accurately predict the price. With better price suggestion estimates, Airbnb home providers can reach an equilibrium price that optimizes profit and affordability. Predicting if a listing has a high/low rating helps the hosts to know if their property is good and how their listing compares to other similar listings. It helps them to make simple changes to the properties they are listing to boost customer satisfaction.

## References

[1] Kalehbasti, P. R., Nikolenko, L., & Rezaei, H. (2019). Airbnb Price Prediction Using Machine Learning and Sentiment Analysis. *ArXiv*. Published. https://arxiv.org/pdf/1907.12665.pdf

[2] Choudhary, P., Jain, A., & Baijal, R. (2018). Unravelling Airbnb Predicting Price for New Listing. *ArXiv*. Published. https://arxiv.org/pdf/1805.12101.pdf

[3] Luo, Y. (2018). What Airbnb Reviews can Tell us? An Advanced Latent Aspect Rating Analysis Approach. *Graduate Theses And Dissertations, Iowa State University*. Published.https://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=7410&context=etd

[4] Dataset links:
https://public.opendatasoft.com/explore/dataset/airbnb-listings/table/?disjunctive.host_verifications&disjunctive.amenities&disjunctive.features
https://www.kaggle.com/samyukthamurali/airbnb-ratings-dataset?select=airbnb-reviews.csv

[5] GitHub link: https://github.com/SYSavy/CIS-5560