JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Comparative analysis of classification algorithms for the hospitality industry.

Submitted to

Amity University Uttar Pradesh

Himanshu Kumar Singh

under the guidance of

Dr. Shailendra Narayan Singh

(Assistant Professor)

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY UTTAR PRADESH

NOIDA (U.P.)

2022

Abstract: Machine learning offers leverage for creating appropriate predictions that outline the domain's decisive frameworks in a world driven by data and decision-making predicted by the captured data of the domain. The consumer and the decision they make that leads to a successful operation or a cancellation are the lifeblood of the hotel sector. In this work, a number of machine learning classification algorithms were implemented in order to compare their performance metrics and identify the necessary features. The hotel business was chosen since it is the most lucrative and fast-paced in order to collect data from past operations and other pertinent factors to determine if a reservation will be successful or canceled.

Keywords: Exploratory Data Analysis, Gradient Descent, Random Forest, XGBoost, LGBM, Adaboost, Accuracy, Precision, Recall, F1-Score, and Support

Introduction

The hotel business was selected as the study's target in order to compare the various machine learning classification methods for the data that was collected. Due to a low client volume and a high rate of reservations that were converted into cancellations during the "COVID-19" pandemic, this industry suffered the greatest revenue losses in the fiscal years 2020 and 2021. The data was collected from the top hotel chains, both domestic and foreign, and delivered and encoded as categorical data for the classification problem in order to produce a better estimate for the volume of reservations based on past customer data among other characteristics (Deloitte Global).

By using classification algorithms, machine learning offers useful methods for resolving this categorization issue. With the use of this information, a trainable categorical dataset was created, which was then utilised to train a variety of machine learning models based on classification methods (Mahesh). On the basis of the availability of

performance measurements including accuracy, F1-Score, recall, and precision, the effectiveness of various models was measured. Depending on the predictions it makes, choose and utilise the model that performs the best.

Correlation Analysis

Using their correlation, it is possible to enlarge on the relationship between the variables in the data. It is the measurement of the magnitude of one variable's change in relation to another variable as dependency or vice versa on a scale from 1 to -1, with 1 depicting a positive correlation as the variance is observed between the variables in the unity and as -1 depicts a negative correlation with the variable variating in the opposite direction, the correlation is scaled at 0, there is no monotonic association between the variables (Senthilnathan) (Hauke and Kossowski). The purpose of the correlation analysis is to describe the relationship between the features that are present in the data, choosing those with a strong positive correlation to create a feature set (Bravais).

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

Equation 1: Pearson Coefficient

x_i: x-variable values in the sample.

$\bar{x}$: mean of values of the x-variable.

y_i: y-variable values in the sample.

$\bar{y}$: mean of values of the x-variable.

Utilizing the Pearson Product-Moment Correlation, heatmap was generated in order to visualize the correlation between the variables and to choose viable features for the feature set as shown below

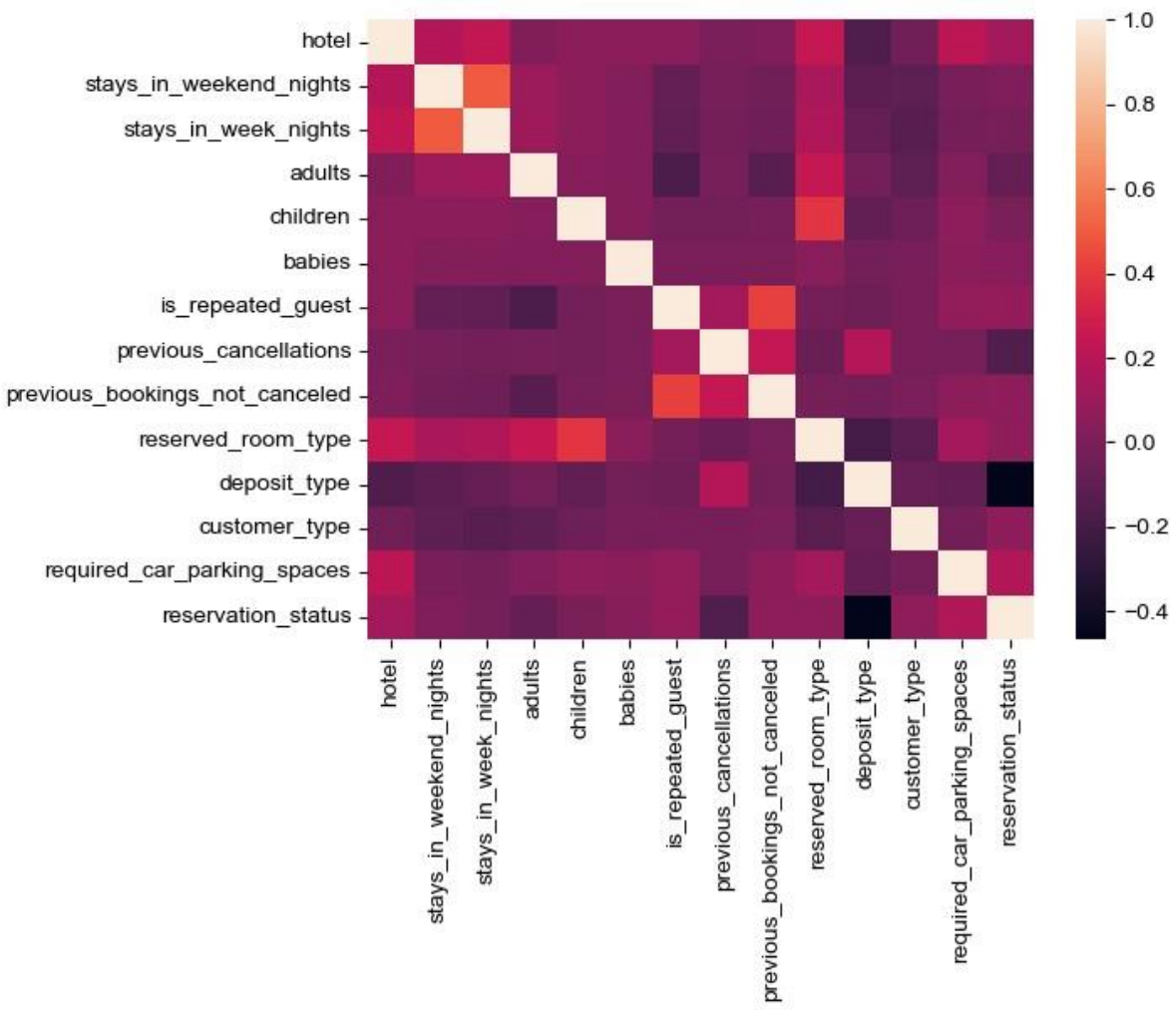


Figure 1: Heatmap visualization for the attributes using Pearson Coefficient

The following credits were used to organize the list of capabilities: "inn," "stays_in_weekend_nights," "stays_in_weekend_nights," "grown-ups," "youngsters," "children," "is_repeated_guest," "previous_cancellations," "previous_bookings_not_canceled," "reserved_room_type," "deposit_type," "customer_type," and "required.

Description of the investigated algorithms

The following algorithms were used for this study, and their performance metrics over the curated dataset were used to compare them.

1. Decision Tree Classifier
2. Random Forest Classifier
3. XGBoost
4. LGBM
5. AdaBoost

Decision Tree Classifier

Similar to how a tree is seen in nature with branches, leaves, and roots, a decision tree has a structure that is made up of nodes rather of actual physical entities called "root nodes," "branches," and "leaf nodes." The generation of the leaf node or multiple leaves as in the case of multivariate values, with designated labels for each leaf node, is achieved by implicitly computing the attribute division at each split level as the split is observed over the node to produce a branch along with a class or a categorical label.

The decision tree's capacity to identify the dataset's most biased feature and its comprehensibility make it advantageous because performance is unaffected by the algorithm's non-linear flow. The splitting criterion must be the same for all nodes

in order for the attributes used to divide to test at any node to identify the "Best" splitting in each class to result in "Pure" branching (Patel and Prajapati).

Here, the decision tree method was used with the following two classification criteria:

Gini Index

The Gini Impurity, also known as the Gini Index, calculates the probability categorization of features that are improperly described when randomly chosen. The Gini Index scales between 0 and 1, with 0 representing pure categorization and 1 representing a random distribution of samples among the classes. The CART, Classification and Regression Tree algorithm, uses the Gini Index.

$$Gini\ Index = 1 - \sum_{i=1}^{n} (P_i)^2$$

Equation 2: Gini Index

P_i: Probability of a sample to be classified among classes.

Information gain

The feature that provides the most information about the classification based on entropy, uncertainty, disorder, or impurity is chosen using the information Gain. From the root node to the leaf nodes, entropy is decreasing.

$$Entropy = -\sum_{i=1}^{n} p_{i*log_2(p_i)}$$

Equation 3: Information gain

p_i: Probability of being a function of entropy.

Random Forest Classifier

It is a functional gone with out of various exclusively ensembled choice trees. Each tree or unit is capable of accurately predicting a class on its own, in accordance with the principle of the wisdom of the crowd. As the primary criterion for distinguishing particular forest trees, each feature can perform individual categorization. In the criteria for the characteristics, the Node impurity reduction is weighted according to its likelihood of reaching the node; the higher the value, the more desirable that characteristic is likely to be (Ali). determining each node's significance using the binary classification assumption.

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

Equation 3: Gini Importance

ni : Priority of node j. j

w : Number of Weighted samples at node j. j

C : Impurity generated at node j. j

$left(j)$: Left split on node j, generating child node.

$right(j)$: Right split on node j, generating child node.

The feature importance is computed as,

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$$

Equation 4: Feature Importance

fi_i: Importance of feature i.

ni_j: Importance of node j.

Later, the feature importance is normalized on the scale of 0 and 1 as,

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j}$$

Equation 5: Normalization function for feature importance

The average of overall trees is computed as,

$$RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T}$$

Equation 6: Final feature importance

$RFfi_i$: The importance of feature i computed from all the trees available in Random Forest Model.

$normfi_i$: Normalized feature importance for i in tree j.

T: Total number of trees.

XGBoost

Extreme Gradient Boosting, or XGBoost, is a different ensemble machine learning approach that uses Decision Trees to achieve gradient boosting for prediction problems. It consists of the gradient-boosted decision trees' distribution. By successively creating weak feature models and adding them while formalising them as gradient descent algorithms rather than utilising an objective function, the gradient boosting accomplishes the boosting and lays the groundwork for the subsequent model (xgboost developers).

LGBM

LightGBM or LGBM is a system in view of slope helping and choice trees to diminish the memory utilization as well as increment the effectiveness of the model.

Gradient-based One Side Sampling

Despite the fact that there is no native, tend tight for data instances in GBDT, it has been observed that data instances with distinct gradients perform distinct functions in the calculation of data gain. In accordance with the definition of information gain, instances with large gradients (i.e., instances that are undertrained) may significantly contribute to data gain. Therefore, for accuracy retention of information gain estimation, once down sampling of the instances. When down sampling, both instances with large gradients that are greater than the threshold and instances with low gradients are rejected. When the value of the data gain includes a large variety, this method provides a better gain estimation than uniformly random sampling with an equivalent or similar target sampling rate.

Exclusive Feature Bundling

In real programs, there are usually a lot of options, but the feature house is usually spread out, giving planners a break and a pretty easy way to cut down on the number of useful features.

AdaBoost

AdaBoost, or adaptive boosting, is a machine learning ensemble technique that builds on decision trees by using weak learners or decision stumps, which are individual split decision trees. It then combines these weak learners or weak classifiers to create a single, united strong classifier in order to classify each class for the samples that are provided (Wang).

Evaluation Metrics

For the need to evaluate the performance of implemented models training there are several parameters available known as performance and quality metrics. These provide with the comparative insight of the performance of these machine learning algorithms over the curated training set of approximately 83500 samples for the selected feature set. These samples underwent data wrangling and label encoding before being used for the training set. The performance metrics as training time and quality metrics as accuracy, precision, weighted mean recall and F1-Score were used. These are operated explicitly from the number of samples in the set

Training Time

The time required for data splitting, data pre-processing, and model evaluation is not included in the time taken by the model to successfully train on the dataset.

Accuracy

It is a metric that shows how well the classification was predicted by the classifier.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ number\ of\ samples}$$

Equation 7: Accuracy

Precision

It is a measurement portraying as the real anticipated cases to be positive. The precision of the classifier's positive response percentage is described.

$$Precision = \frac{True\ \ Positives}{True\ Positives + False\ Positives}$$

Equation 8: Precision

Recall

It is a metric for evaluating the accuracy of the model's predictions regarding actual positive cases.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Equation 9: Recall

F1-Score

It gives the consolidated thought regarding the Accuracy and review Measurements.

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

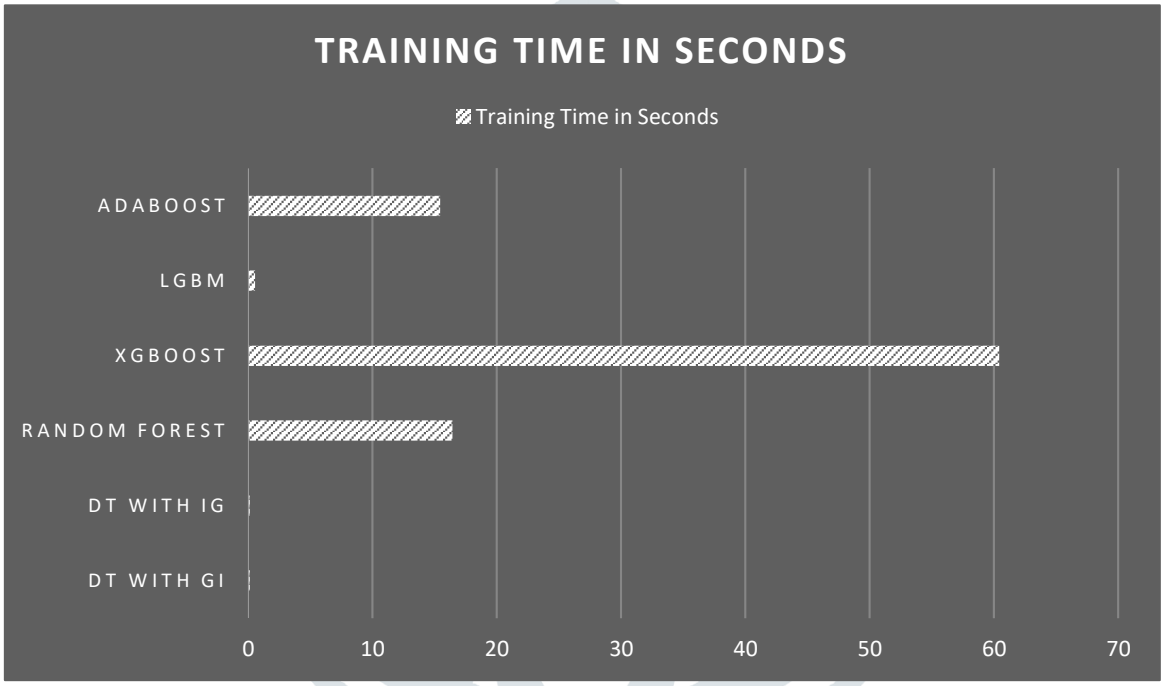
Equation 10: F1-Score

Comparative Analysis

As shown in the following table 1, training time was recorded following the thorough analysis and training of the aforementioned algorithms to determine how much time each machine learning model needed to train over the training set.

S No.	Algorithm	Training time in seconds
1	Decision Tree using Gini Index	0.10481
2	Decision Tree using Information Gain	0.09888
3	Random Forest Ensemble	16.42484
4	XGBoost	60.41680
5	LGBM	0.52373
6	AdaBoost	15.43733

Table 1: Training Time for the models.



As depicted by the figure 2, XGBoost tend to be the slowest in terms of the training time, because of the extreme gradient boosting at its core.

The accuracy of these models in predicting the right categorization from the test set is shown in table 2 below.

S No.	Algorithm	Accuracy in percentage
1	Decision Tree using Gini Index	76. 66685
2	Decision Tree using Information Gain	76. 65848
3	Random Forest Ensemble	76. 99073
4	XGBoost	77. 21130
5	LGBM	77. 43466
6	AdaBoost	77. 14429

Table 2: Prediction accuracy for the models.

This suggests that all the models function similarly accurately in the vicinity of one another at 77%, with a variance of roughly 0.28%, and that the LGBM performs the best. About the other metrics, the following are presented in the tables below as a classification report in percentage for their classification for the goal variable

as "reservation status," with classifications being "Cancel," "Stay," and "No-Show" for the models that were implemented.

Labels	Precision	Recall	F1-Score	Support
Cancel	86	45	59	12921
Stay	75	96	84	22576
No-Show	23	2	3	319

Table 3: Classification Report for Decision Tree with Gini Index.

Labels	Precision	Recall	F1-Score	Support
Cancel	86	45	59	12921
Stay	75	96	84	22576
No-Show	15	2	3	319

Table 4: Classification Report for Decision Tree with Information Gain.

Labels	Precision	Recall	F1-Score	Support
Cancel	88	44	59	12921
Stay	75	97	84	22576
No-Show	29	2	4	319

Table 5: Classification Report for Random Forest.

Labels	Precision	Recall	F1-Score	Support
Cancel	90	44	59	12921
Stay	74	97	84	22576
No-Show	42	2	3	319

Table 6: Classification Report for XGBoost.

Labels	Precision	Recall	F1-Score	Support
Cancel	95	42	58	12921
Stay	74	99	85	22576
No-Show	22	1	1	319

Table 7: Classification Report for LGBM.

Labels	Precision	Recall	F1-Score	Support
Cancel	95	41	57	12921
Stay	74	99	85	22576
No-Show	0	0	0	319

Table 8: Classification Report for AdaBoost.

Conclusion

When all the algorithms used on the dataset were successfully implemented, it was clear that the Decision Tree was trained on the data the quickest compared to the other algorithms, even though it was the least accurate and only came close to the mean accuracy of 77.02%. The most accurate model, which was based on LGBM, also performed similarly to the Decision Tree in terms of training time. When compared amongst the various machine learning methods employed, the other evaluation criteria remained constant. This shows that regardless of the classification algorithm used, the quality of predictions depends on the training data.