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.

Ashu, Dr. Leena Singh

Amity University

Abstract *– In a world driven by data and decision-making predicted by the captured data of the domain, machine learning provides leverage for making suitable predictions that frame the domain's decisive frameworks. The hospitality industry survives on the customer and the choice that it opts for resulting in a successful operation or a cancellation. In this work, the implementation of various machine learning classification algorithms was performed for the comparative analysis of their performance metrics as well as to discover the necessary features. The hotel industry is the most lucrative and fast-paced and was chosen for data capturing through their history as well as other relevant parameters, predicting a successful operation conversion for predicting a reservation to be successful or cancellation.*

Keywords – Machine Learning; Exploratory Data Analysis; Decision Tree; Random Forest; XGBoost; Gradient Descent; LGBM; Adaboost; Accuracy; Precision; Recall; F1-Score; Support

Introduction

In the provided work the Hotel industry was chosen as the beneficiary of the study to provide a comparative analysis of the available machine learning classification algorithms for the captured data. During the pandemic of "COVID-19", the most affected industry was this, resulting in massive amounts of revenue losses during the fiscal years of 2020 and 2021 due to low customer volume and a high amount of conversion of reservations into cancellations. To provide a better forecast for the volume of reservations based on previous customer data among various parameters, the data was captured from the leading hotel chains both domestic as well as international, provided and encoded as categorical data for the classification problem (Deloitte Global).

Machine Learning provides substantial techniques for solving this classification problem by the means of classification algorithms. This data was used to curate a trainable categorical dataset for the subsequent training of various machine-learning models based on the classification algorithms (Mahesh). The performance of various models was metered on the availability of the performance metrics such as accuracy, F1-Score, recall and precision. Select and leverage the best-performing model based on the predictions made by it.

Correlation Analysis

The association between the variables in the data could be elaborated using the correlation between them. It is the measurement of the change in the magnitude of a variable with respect to another variable as dependency or vice versa, on a scale of 1 to -1, with 1 depicting a positive correlation as the variance is observed between the variables in the unity whereas as -1 depicts a negative correlation with the variable variating in the opposite direction, the correlation is scaled at 0, there exists no monotonic association between the variables (Senthilnathan) (Hauke and Kossowski). The aim of the correlation analysis is to provide with a description of the correlation between the available attributes in the data, selecting the attributes with a high positive correlation to curate a feature set (Bravais). A

requisite to curate a feature set for the experimental analysis, the Pearson Product-Moment Correlation was analysed (Rousseau, Egghe and Guns), providing the scalable measurement of a linear association among two variables as correlation coefficient

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

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

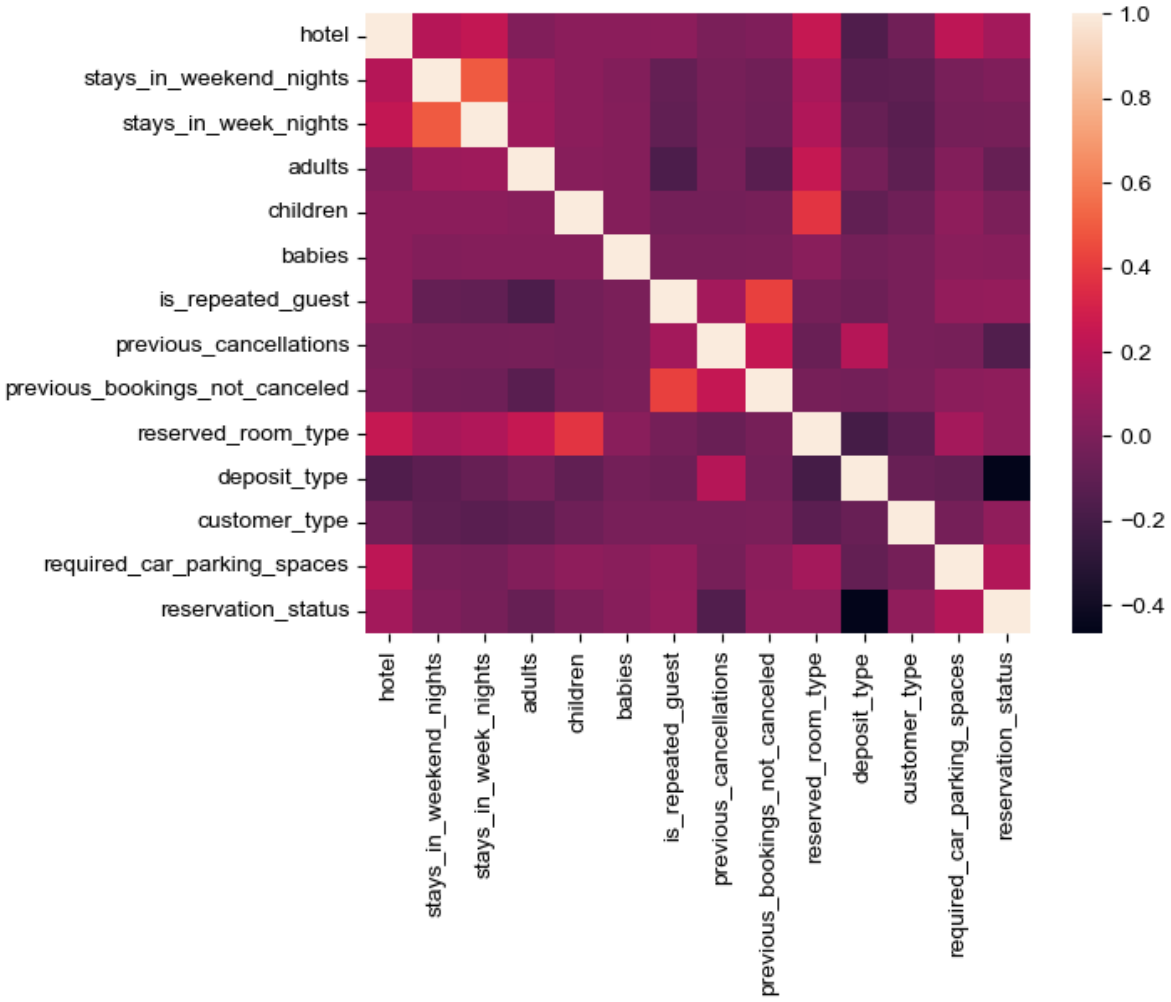

Figure 1: Heatmap visualization for the attributes using Pearson Coefficient

Upon investigation of the heatmap, the following attributes were used to curate the feature set as, “hotel”, “stays_in_weekend_nights”, “stays_in_week_nights”, “adults”, “children”, “babies”, “is_repeated_guest”, “previous_cancellations”, “previous_bookings_not_canceled”, “reserved_room_type”, “deposit_type”, “customer_type”, “required_car_parking_spaces”, followed by the target variable which would be as “reservation_status” providing the booking made will be successful an operation or will be cancelled.

Description of the investigated algorithms

For the conduct of this study the following algorithms were implemented compared on the basis of their performance metrics over the curated dataset:

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

Decision Tree Classifier

As similarly as a tree is observed in nature with branches, leaves and roots, decision tree follows with the same structure, consisting of nodes instead of physical entities naming as "*root node*", "*branches*" and "*leaf nodes*". Implicitly computing the attribute division at each split level, as the split is observed over the node to generate a branch along with class or the categorical label, resulting in the generation of the leaf node or multiple leaves as for the case of multivariate values, with designated labels for each leaf node. The advantage that the decision tree provides is with the ability of selection for the most biased feature in the dataset and the comprehensibility nature, performance not being affected by the non-linearity flow of algorithm. The attributes used to split to test at any node in order the determine the 'Best' splitting in individual classes, resulting in the branching to be as 'Pure', provided the splitting criteria has to be identical (Patel and Prajapati).

The decision tree algorithm was utilised here with two classification criterions are followings:

Gini Index

Gini index or the Gini Impurity computes the probabilistic classification of feature that is incorrectly specified upon selected randomly. The scalability of the Gini Index varies between 0 and 1, pure classification being expressed by the 0 whereas 1 point towards the random distribution of the samples across the classes. The Gini Index is deployed in the CART, Classification and Regression Tree algorithm.

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

Equation 2: Gini Index

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

Information gain

The information Gain is used for the selection of the feature providing the maximal information about the classification based upon the entropy, uncertainty, disorder or impurity is quantified. The entropy is descending from the root node to the leaves nodes.

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

Equation 3: Information gain

p_i: Probability of being a function of entropy.

Random Forest Classifier

It is an operative consisting of multiple ensembled individual decision trees. Each tree or a unit is capable of individually predicting a class, following the persona of wisdom of crowds. Each feature is capable of performing individual classification as primary splitting criteria for individual trees in the forest. The criteria used for the features is, feature importance, calculated as the reduction in the Node impurity is weighted according to its chances of getting to the node., and the higher the value, the more desired that characteristic is likely to be (Ali). Calculating each node's relevance under the assumption of binary classification

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

Equation 4: Gini Importance

ni_j: Priority of node *j*.

w_j: Number of Weighted samples at node *j*.

C_j: Impurity generated at node *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 5: 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 6: 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 7: 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

XGBoost is another ensemble Machine Learning algorithm implementing the Decision Trees in accordance with gradient boosting for the prediction problems, standing for 'Extreme Gradient Boosting'. It consists of the distribution of the gradient-boosted decision trees. The gradient boosting implements the boosting by consecutively generating weak feature models and adding them, formalizing them as gradient descent algorithm rather than using a objective function, providing a foundation for the next model (xgboost developers).

LGBM

LightGBM or LGBM is a framework based on gradient boosting and decision trees to decrease the memory usage as well as increase the efficiency of the model.

Gradient-based One Side Sampling

While there's no native, tend tight for data instance in GBDT, It is noticed that data instances with totally different gradients play different roles within the computation of data gain. In particular, in step with the definition of information gain, those instances with giant gradients (i.e., under-trained instances) can contribute a lot of to the data gain. Therefore, once down sampling of the info instances, for accuracy retention of information gain estimation. Upon down sampling, acceptance of instances with the large gradients, greater than threshold as well rejection of the low gradient instances is observed. This process provides with a better gain estimation as compared with the uniformly

random sampling, with an equivalent or similar, target sampling rate, particularly once the worth of data gain includes a large variety.

Exclusive Feature Bundling

Although there are typically a lot of options in real programmes, the feature house is usually somewhat spread, giving planners a breather and a practically lossless way to reduce the amount of useful features.

AdaBoost

Adaptive Boosting also known as AdaBoost is an ensemble method in machine learning building upon the decision trees, as utilising individual split decision trees known as weak learners or decision stumps. It then combines that weak learners or weak classifiers together to form a united single strong classifier, as for every class to be classified for the available samples (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

It is referred to as the taken by the model to train on the dataset successfully, it does not include the time taken for the data splitting, data pre-processing and model evaluation.

Accuracy

It is a metric which provides with the measure of the classifier correctly predicting the classification.

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

Equation 8: Accuracy

Precision

It is a metric describing as the actual predicted cases to be positive. Precision describes what percentage of positive answers from the classifier are correct.

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

Equation 9: Precision

Recall

It is a metric to measure the prediction of the model to predict actual positive cases.

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

Equation 10: Recall

F1-Score

It provides the combined idea about the Precision and recall Metrics.

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

Equation 11: F1-Score

Comparative Analysis

Upon the thorough analysis and training of the above-mentioned algorithms, training time was documented in order to evaluate how much each of the machine learning model took time to train over the training set, as shown in the following table 1,

S No.	Algorithm	Training time in seconds
1	Decision Tree using Gini Index	0.10482
2	Decision Tree using Information Gain	0.09889
3	Random Forest Ensemble	16.42485
4	XGBoost	60.41681
5	LGBM	0.52374
6	AdaBoost	15.43734

Table 1: Training Time for the models.

From the above provided data it could be easily inferenced that the conventional Decision tree was the fastest model to be trained while using information gain as the criteria, this could be easily visualized as the following figure 2 depicts,

Figure 2: Comparative chart for the training time.

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 following table 2 depicts the accuracy of these models predicting the correct classification from the test set,

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.

From this it could be inferenced that all the models perform with similar accuracy in the neighbourhood of each other at 77%, deviation of about 0.28% with the best performing model being the LGBM. As the other metrics are concerned the following are provided in the following tables as classification report in percentage for their classification for the target variable as 'reservation_status' and classifications being 'Cancel', 'Stay' and 'No-Show' for the implemented algorithms as models,

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

Upon the successful implementation of all the algorithms used on the dataset, it could be easily inferenced that the Decision Tree trained on the data fastest as compared to rest algorithms while being least accurate but being in the neighbourhood of the mean accuracy of the 77.02%, while the most accurate model being based of LGBM as well performing in relevance to the Decision Tree in terms of training time. The rest of the evaluation metrics stayed similar when compared across the machine learning algorithms used. This depicts that the quality of predictions made are dependent of the training data irrespective of the classification algorithm used.

References

Ali. "Random Forests and Decision Trees." *International Journal of Computer Science Issues* (2012).

Bravais, A. “Analyse mathématique sur les probabilités des erreurs de situation d'un point.” *Memoires Par Divers Savan* (2022): 255-332.

Deloitte Global. *Impact of COVID-19 on the hospitality industry*. 2021. <https://www2.deloitte.com/nl/nl/pages/consumer/articles/impact-of-covid-19-on-the-hospitality-industry.html>.

Hauke, Jan and Tomasz Kossowski. “Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data.” *Quaestiones Geographicae* (2011): 87-93.

Mahesh, Batta. “Machine Learning Algorithms - A Review.” *International Journal of Science and Research (IJSR)* (2018).

Microsoft Corporation. *LightGBM Documentation*. October 2022. <https://lightgbm.readthedocs.io/en/v3.3.2/Parallel-Learning-Guide.html>.

Patel, Harsh H and Purvi Prajapati. “Study and Analysis of Decision Tree Based Classification Algorithms.” *International Journal of Computer Sciences and Engineering* (2018).

Rousseau, Ronald , Leo Egghe and Raf Guns. “Chapter 4 - Statistics.” Rousseau, Ronald, Leo Egghe and Raf Guns. *Becoming Metric-Wise*. Chandos Publishing, 2018. 67-97.

Senthilnathan, Samithamby. “Usefulness of Correlation Analysis.” *SSRN Electronic Journal* (2019).

Wang, Ruihu. “AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review.” *Physics Procedia* (2012): 800-807.

xgboost developers. *XGBoost Documentation*. October 2022. <https://xgboost.readthedocs.io/en/stable/>.