**Predicting Hotel Reservation Rebooking with Multiple Logistic Regression**

Heather Marie Rauch

College of Information Technology Western Governors University

Daniel J. Smith, Ph.D., MBA

February 3, 2023

# Predicting Hotel Reservation Rebooking with Multiple Logistic Regression

In the hospitality industry, it is common for customers to cancel and rebook a hotel reservation when they find a better rate online for the same package. These actions have a two-fold impact on the hotel business. First, there is lost revenue equal to the difference in room rates. Second, metrics used to forecast room demand are inflated by repeat bookings (Clay, 2023). There is plenty of research on the prediction of hotel reservation cancelations (Putro, Septian, Widiastuti, Maulidah, & Pardede, 2021) (Antonio, Almeida, & Nunes, 2017) (Sanchez-Medina & C.-Sancheq, 2020). There is not, however, much information on rebooking. For this reason, this analysis aims to answer the research question, can hotel reservation rebooking be modeled with Multiple Logistic Regression using the Hotel Reservations Dataset?

Using Multiple Logistic Regression, this analysis will attempt to identify, and model canceled and rebooked reservations. Multiple Logistic Regression uses two or more independent features of the data, in this case, the hotel reservations, to predict the probability of a dichotomous, outcome (Boston University School of Public Health, 2013). For this analysis, the dichotomous outcome will be either 1, representing a canceled and rebooked reservation, or 0, representing a reservation that was kept or canceled and not rebooked. The alternative hypothesis, shown below with the null hypothesis, attempts to identify which features of the Hotel Reservations Dataset will be statistically significant in predicting whether a reservation will be rebooked.

H0: The booking lead time and average price per room will not statistically significantly predict that a reservation will be rebooked.

Ha: The booking lead time and average price per room will statistically significantly predict that a reservation will be rebooked.

# ****Data Collection****

To perform this analysis, hotel reservation records were required. To meet this need, the Hotel Reservations Dataset from Kaggle, made publicly available under the creative commons license, was used (Raza, 2023). This dataset contained 19 columns and 36,275 rows, each of them about a single hotel reservation. The included variables can be seen in Table 1.

**Table 1**

*Variables Included in the Hotel Reservations Dataset*

|  |  |  |
| --- | --- | --- |
| Field | Type | Dependency |
| no\_of\_adults | Continuous | Independent |
| no\_of\_children | Continuous | Independent |
| no\_of\_weekend\_nights | Continuous | Independent |
| no\_of\_week\_nights | Continuous | Independent |
| type\_of\_meal\_plan | Categorical | Independent |
| required\_car\_parking\_space | Categorical | Independent |
| room\_type\_reserved | Categorical | Independent |
| lead\_time | Continuous | Independent |
| arrival\_year | Continuous | Independent |
| arrival\_month | Continuous | Independent |
| arrival\_date | Continuous | Independent |
| market\_segment\_type | Categorical | Independent |
| repeated\_guest | Categorical | Independent |
| no\_of\_previous\_cancellations | Continuous | Independent |
| no\_of\_previous\_bookings\_not\_canceled | Continuous | Independent |
| avg\_price\_per\_room | Continuous | Independent |
| no\_of\_special\_requests | Continuous | Independent |
| booking status | Categorical | Independent |

The biggest advantage of using prepared data from Kaggle was that there was generally less data preparation required. The data were already tidy, meaning they were organized in rows and columns, and there were no duplicate records or missing fields. This allowed more time to be spent on data wrangling and analysis tasks. The biggest disadvantage of using prepared data from Kaggle was that the analysis was limited to variables included in the data set. If the data were extracted directly from a hotel reservation database, it would have taken more work to clean and organize, but there would have been a larger selection of variables that could have been considered for inclusion in the final model. Direct extraction from a database would also have allowed the inclusion of information needed to link rebooked reservations to their initial cancellations. According to the author of this dataset, it is “training example data” provided within their platform (Raza, 2023). Because of this, there was no such identifying information. This was the biggest challenge in the data collection process. To overcome this challenge, deterministic matching was utilized. Deterministic matching is used to identify “exact matches” through a rule-based approach (Chandramohan, 2020).

# ****Data Extraction and Preparation****

To prepare the data for analysis, Python 3.9 was used. Python is one of the fastest-growing general-purpose programming languages available today. It is widely used in the data science and analytics fields because it is easy to learn, free to use, has a wide user community, and there exists an entire ecosystem of data management, statistics, and data science libraries (chandan07, 2022). One of the biggest advantages of using Python for data preparation is the pandas library (The pandas development team, 2022). The pandas Python library offers a multitude of tools for data extraction and preparation from directly querying relational databases, to extracting JSON from web requests to pulling in comma-delimited files. Additionally, it includes an entire suite of methods designed to operate on data the way one would with Structured Query Languages (SQL). The biggest disadvantages associated with using Python for data extraction and preparation are that it is not as fast as other languages like Java, C, and C++ due to its interpreted nature, and it tends to consume more memory because of its dynamic typing (vartika02, 2021).

To prepare the data for analysis, there were several steps taken. After loading the data from a comma-delimited file into a pandas DataFrame, the data structures were reviewed. The initial data consisted of 36,275 rows and 19 variables.

**Figure 1**

*Screenshot: Data Structures*

Table

Description automatically generated

The phenomenon of rebooking the same hotel room for a lower rate occurs online (Clay, 2023). Before taking any additional steps, the data was filtered to only include reservations that were booked online. The remaining 23,214 rows were inspected for missing values and duplicate rows.

**Figure 2**

*Screenshot: Values Remaining After Applying Online Filter*



**Figure 3**

*Screenshot: Duplicates and Missing Values*

Table

Description automatically generated

Next, the presence of outliers was investigated. Using the SciPy library (Virtanen, et al., 2020), z-scores were calculated for each variable. Any value with a z-score above 3 or below -3 was considered an outlier. Because regression analysis is sensitive to outliers (Boston University School of Public Health, 2013), they were all excluded from the analysis. This left a total of 21, 221 rows for the analysis.

**Figure 4**

*Screenshot: Outliers*

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 5**

*Screenshot: Rows Remaining After Outlier Removal*



To identify the target variable, rebooked reservations, deterministic matching was used (Chandramohan, 2020). The following ruleset was used to match canceled reservations to the subsequently rebooked reservation:

1. The first booking’s status must be Canceled.
2. The second booking’s status must be Not Canceled.
3. The following variables match exactly between both reservations: arrival date, room type reserved, type of meal plan, required car parking space, number of adults, number of children, number of weekend nights, and number of weeknights.
4. The first booking’s average price per room must be greater than the second booking’s average price per room.
5. The first booking’s lead time must be greater than the second booking’s lead time.

The goal of this rule set was to identify canceled hotel reservations where the same package was rebooked for the same check-in date at a lower rate. Canceled reservations that were successfully matched to a booked reservation were assigned the rebook label of 1. All other records were assigned the rebook label of 0. This resulted in 998 rebooked reservations and 20,223 canceled or kept reservations.

**Figure 6**

*Screenshot: Rebook Value Counts*

Table

Description automatically generated with low confidence

With the target variable identified, the numeric variables were scaled using the Scikit-learn Python library (Pedregosa, et al., 2011). Numeric scaling is necessary for regression analysis to prevent variables measured on a large scale from outweighing variables measured on a small scale during the fitting process (Boston University School of Public Health, 2013).

**Figure 7**

*Screenshot: Numeric Variables Scaled*

Table

Description automatically generated

Next, the categorical variables were one-hot encoded. This created one column for each possible category and filled them with a 1 if the record was in the category and a 0 if it was not. One category was dropped for each of the categorical fields to prevent multicollinearity in the final model (JMP Statistical Discovery LLC, 2023).

**Figure 8**

*Screenshot: Categorical Variables One-hot Encoded*

Table

Description automatically generated with medium confidence

The prepared data was assessed for multicollinearity using SciPy’s variance inflation factor (VIF) function (Virtanen, et al., 2020). A VIF value of over 5 indicates there is multicollinearity present (The Investopedia Team, 2022).

**Figure 9**

*Screenshot: Initial Variance Inflation Factor*

Table

Description automatically generated

After repeated\_guest\_1 was removed, type\_of\_meal\_plan\_Meal\_Plan\_1 still had a VIF value over 5. After removing type\_of\_meal\_plan\_Meal\_Plan\_1, all values were within the acceptable range.

**Figure 10**

*Screenshot: Final Variance Inflation Factor*

Table

Description automatically generated

The final preparatory step taken was splitting the data. Using the Scikit Learn Python library function (Pedregosa, et al., 2011), train\_test\_split, the data was split so that 20% was reserved for testing. The split was stratified by the target variable to ensure the two sets remained comparable.

**Figure 11**

*Screenshot: Shape of Training and Testing Data*

Text

Description automatically generated

# ****Analysis****

The data analysis was performed in four stages: univariate analysis, bivariate analysis, model development, and post hoc model development.

## Univariate Analysis

During the univariate analysis stage, each variable was visualized using the Matplotlib Python library (Hunter, 2007). Numeric variables were displayed using histograms and categorical variables were displayed using bar charts. The visuals in Figure 12 were produced after removing outliers, but before scaling numeric variables or one-hot encoding categorical variables.

**Figure 12**

*Univariate Analysis*

Diagram

Description automatically generated

Looking at the numeric variables, the no\_of\_adults variable contains three unique values ranging from 1 to 3 that are normally distributed. The no\_of\_children variable contains two unique values and is skewed positively with the majority of reservations including no children. The no\_of\_weekend\_nights variable ranges from 0 to 3 and is positively skewed, with the majority of reservations including no weekend nights. The no\_of\_week\_nights variable ranges from 0 to 6 and is normally distributed except for a slight positive skew. The lead\_time variable is very positively skewed with the majority of reservations booked 50 or fewer days before the arrival date. The arrival\_year variable contains two values 2017 and 2018, with the majority of the reservations occurring in 2018. The arrival\_month variable is negatively skewed indicating that the majority of reservations occur closer to the end of the year. The arrival date variable is nearly uniform, indicating that reservations are evenly booked throughout each month. The no\_of\_previous\_cancellations variable ranges from 0 to 1 with the vast majority of reservations having 0 previous cancellations. The same is true for the no\_of\_previous\_bookings\_not\_canceled variable. The avg\_price\_per\_room variable is mostly normally distributed with a slight positive skew. The no\_of\_special\_requests variable ranges from 0 to 3 with a positive skew.

Looking at the categorical variables, the type\_of\_meal\_plan variable has three categories with the majority of reservations opting for Meal Plan 1. The required\_car\_parking\_space variable is dichotomous with the majority of reservations not requiring a parking space. The room\_type\_reserved variable has 7 possible options, but most reservations are for room type 1. Because the data was filtered to online reservations, only one category remains for the market\_segment\_type variable. The repeated guest variable is dichotomous with most guests visiting for the first time. The booking\_status variable has two categories with most reservations being Not\_Canceled.

At this point in the analysis, the following variables were dropped: arrival\_year, arrival\_date, and market\_segment\_type. The arrival\_year variable would not have extrapolated well to future reservations, the arrival date had a uniform distribution, and market\_segment\_type only had a single category remaining.

## Bivariate Analysis

The bivariate Analysis was performed in two steps, both using the Matplotlib Python library (Hunter, 2007). First, the correlation matrix was reviewed to get a general idea of the relationships between the independent variables. The strongest correlation, which was still weak, occurred between no\_of\_adults and avg\_price\_per\_room with a value of 0.3.

**Figure 13**

*Correlation Matrix*

A picture containing chart

Description automatically generated

The second step of the bivariate analysis was to visualize each variable against the target variable rebook. Numeric variables were displayed using boxplots and categorical variables were displayed using bar charts.

**Figure 14**

*Bivariate Analysis*

Diagram, engineering drawing

Description automatically generated

There was a noticeable difference in the lead\_time variable between the rebook Yes and No groups with rebooks being booked with more lead time. There was a slight difference between the Yes and No rebook groups for the avg\_price\_per\_room and no\_of\_special\_requests variables. Rebooked reservations appeared to cost more on average and had fewer special requests. It was difficult to tell from a visual analysis of the categorical variables what impact they may have had on the rebook variable because of the difference in volume between the Yes and No groups. However, it appeared that most rebooked reservations opted for Room Type 1 and Meal Plan 1, did not require parking, were not repeated guests, and did cancel their initial reservation.

## Model Development

Before developing the Multiple Logistic Regression model, the prepared data was assessed to ensure none of the model assumptions would be violated.

***Model Assumption 1: Binary Response Variable***

The target variable, rebook, contains two unique values.

**Figure 15**

*Screenshot: Unique Rebook Values*



***Model Assumption 2: Independent Observations***

This assumption was possibly violated by repeat guests who visited the hotel more than one time during the sample period. If this did occur, it was at a low volume.

***Model Assumption 3: No Multicollinearity***

Multicollinearity was assessed and addressed during the data preparation phase.

***Model Assumption 4: No Outliers***

All outliers were removed during the data preparation phase.

***Model Assumption 5: Linearity Between Predictors and Logit of Target***

To test the linearity between each independent variable and the logit of the target variable, the Box-Tidwell test was performed. This tests whether a statistically significant relationship exists between the target variable and the log interactions of the predictors. If found this relationship indicates the absence of linearity between the predictor and the logit of the target (Leung, 2021).

**Figure 16**

*Screenshot: Box-Tidwell Test Results*

Table

Description automatically generated

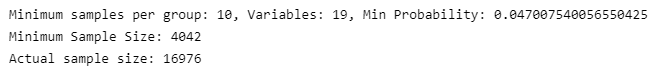
A statistically significant relationship was found between the no\_of\_week\_nights variable’s log interaction and the target variable. This indicates that there is not a linear relationship between the no\_of\_week\_nights variable and the logit of the rebook variable. Before fitting the model, this variable was dropped from the training data.

***Model Assumption 6: Sufficient Sample Size***

In general, to meet this assumption, there must be at least 10 instances of the target with the lowest probability for each predictor. To calculate the minimum sample size required to predict the target, the following formula was used (Bobbitt, 2020).

**Figure 17**

*Screenshot: Sample Size Results*



With all assumptions checked, the initial Multiple Logistic Regression model was built and fitted to the testing data. There were several issues with the initial model. During fitting the model failed to converge, several of the variables were not statistically significant, and quasi-complete separation was detected. Quasi-complete separation can prevent the model from converging and tends to produce biased results (Lu).

**Figure 18**

*Screenshot: Initial Model Summary*

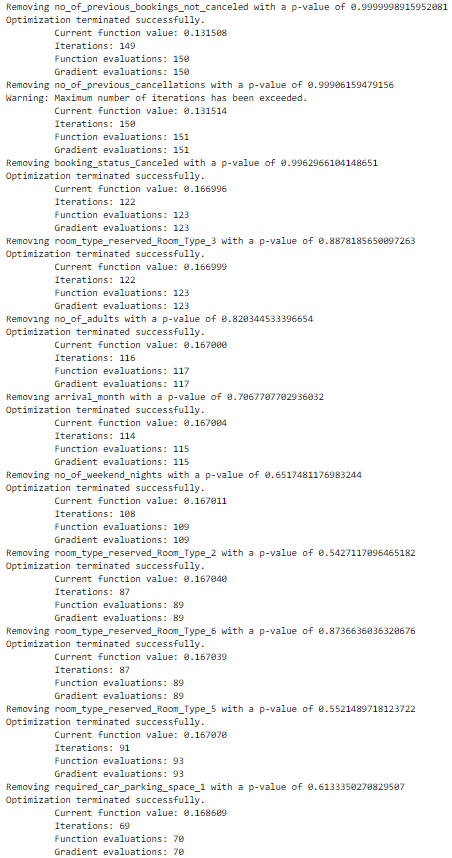
Table

Description automatically generated

To improve the model, backward stepwise feature selection was performed by iteratively removing the variable with the highest p-value until all remaining predictors were statistically significant.

**Figure 19**

*Screenshot: Feature Selection*



The reduced model had a lower Pseudo R-squared value, reducing from 0.3064 for the initial model to 0.1108 for the reduced model. However, the model did successfully converge, all of the predictors were statistically significant, and no warnings were presented.

**Figure 20**

*Screenshot: Reduced Model Summary*

Table

Description automatically generated

To assess the model’s performance, the Accuracy and recall scores were used. Accuracy is important to monitor for the overall model and recall specifically identifies how often the target is correctly identified when it exists (Heeswijk, 2022).

While the reduced model did have an Accuracy of 0.95, it was completely unable to identify rebooked reservations.

**Figure 21**

*Screenshot: Reduced Model Metrics*



## Post-hoc Model Development

During the final stage of analysis, a Feed-Forward Neural Network was created using the TensorFlow Python library (Abadi, et al., 2015). Feedforward Neural Networks pass data through an artificial mesh of hidden layers, processing the features of the data using trainable weights, bias, and activation functions to produce the final output (Kurama & Whitfield, 2022). The final architecture contained an input layer, two dense hidden layers, and an output layer. Both hidden layers utilized the Rectified Linear Unit (reLU) activation function. reLU is a piecewise function that converts negative values to zero. It is commonly used in feedforward networks because it is not subject to the vanishing gradient problem that occurs when using the sigmoid or hyperbolic tangent functions (Brownlee, 2020). The equation for reLU is:

The first hidden layer contained 500 neurons, the second hidden layer contained 250 neurons, and the output layer contained a single neuron activated by the sigmoid function to produce a binary probability distribution. During the compilation of the model, the Adam optimizer was used. This optimizer combines the gradient descent concepts of momentum and root mean square propagation (RMSP) to converge faster and overcome local minima better than other options (prakharr0y, 2020). To measure loss, Balanced Cross-Entropy was used because of the class imbalance present in the data. Recall when verifying the assumptions for Multiple Logistic Regression, the probability of the target outcome, a reservation being rebooked, was less than 5%. Because the target occurs so infrequently in the data, bias was introduced into the regression model. Balanced Cross-Entropy accounts for this by weighting the classes using an alpha parameter (Nayak, 2022). For this model, alpha was set to 0.94, the approximate inverse of the target probability. The Accuracy and Recall scores were again used to measure the model’s performance since the goal was to predict when the target outcome occurs. The resulting network contained 129,501 trainable parameters.

**Figure 22**

*Screenshot: Neural Network Summary*

Table

Description automatically generated

**During the model fitting process, an early stopping monitor was used to prevent overfitting to the training data. The monitored metric was the loss, balanced cross-entropy, on the validation data. The monitor was configured to allow the validation loss not to improve for five epochs before restoring the best weights and terminating the fitting process. This allowed the model to overcome local minima.**

**Figure 23**

***Screenshot: Neural Network Fitting***

**A picture containing text

Description automatically generated**

**Epoch 2 had the lowest validation loss, so the weights from that epoch were restored before terminating the fitting process. The recall and accuracy were not monitored for early stopping because the two metrics fight back and forth as the model loss minimizes, causing erratic and unreliable results. Monitoring the loss results in a balance between Accuracy and recall.**

**Figure 24**

***Neural Network Recall and Loss***

Chart, line chart

Description automatically generated

The metrics using the restored weights can be seen in Figure 25. The training recall and Accuracy were 0.74 and 0.72, respectively. The validation recall and Accuracy were 0.74 and 0.71, respectively. The fact that the scores were all relatively close was a good sign that overfitting was not present and that the model would perform well on out-of-sample predictions.

**Figure 25**

*Screenshot: Neural Network Training and Validation Performance*

**Text

Description automatically generated**

**Finally, predictions were made and evaluated using the reserved testing data. Completely in line with the training and validation data, the testing data produced Accuracy and recall scores of approximately 0.73.**

**Figure 26**

***Screenshot: Neural Network Testing Performance***

****

**Figure 27**

***Neural Network Confusion Matrix***

Chart, treemap chart

Description automatically generated

# ****Data Summary and Implications****

This analysis aimed to determine whether hotel reservation rebooking could be modeled with a Multiple Logistic Regression model using the Hotel Reservations Dataset. The null hypothesis stated that the booking lead time and average price per room would not statistically significantly predict whether a reservation would be rebooked. Conversely, the alternative hypothesis stated that the booking lead time and average price per room would statistically significantly predict whether a reservation would be rebooked. Both lead time and average price per room were found to be statistically significant when attempting to predict whether a hotel reservation would be rebooked using a Multiple Logistic Regression model. Because of this, the null hypothesis can be rejected in favor of the alternative hypothesis. That being said, even though the regression had an Accuracy score of 0.95, the recall was 0, meaning that the model was unable to identify rebooked reservations when they occurred. Using the same statistically significant variables in a feed-forward neural network customized to minimize the balanced cross-entropy was far more capable of identifying whether a reservation would be rebooked with a test recall of 0.73 but at the cost of overall model Accuracy, which dropped to 0.73.

The biggest limitation of this analysis was that the data was extracted from a hospitality training environment with all identifiers removed. Because of this, the links between the initially canceled and rebooked portions of the rebooked reservation had to be estimated using deterministic matching. This means that the neural network that resulted from this analysis can only be considered a proof of concept and cannot be directly applied to a hospitality system’s data with any level of confidence. Because of this, it is recommended that the next course of action be to extract hotel reservation data with confirmed links between the canceled and rebooked portions of the rebooked reservations. That data should then be used to fit the neural network and test the performance to determine if any side effects were introduced by using data extracted from a hospitality training environment. The model resulting from that analysis would be a far better candidate for application in a real hospitality setting. To build upon this analysis the following two projects are proposed:

1. Exclude the initially canceled portion of the deterministically matched rebooked reservations and use the data to produce a time-series model of hotel demand. Compare how the results differ when the time-series model is applied to the same data but with the predicted rebooks removed instead of the deterministically matched rebooks.
2. Analyze the seasonality of rebooked reservations and average price per room to determine if any pattern exists between the time of year, how the price fluctuates, and whether a reservation will be rebooked.

# References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., . . . Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* doi:<https://zenodo.org/record/6574269>

Antonio, N., Almeida, A., & Nunes, L. (2017, December 1). *[PDF] Predicting Hotel Bookings Cancellations with a Machine Learning Classification Model*. Retrieved from Semantic Scholar: <https://www.semanticscholar.org/paper/Predicting-Hotel-Bookings-Cancellation-with-a-Model-Antonio-Almeida/38fb00e49a34a15f692add91e889a51a3b994904>

Bobbitt, Z. (2020, October 13). *The 6 Assumptions of Logistic Regression (With Examples)*. Retrieved from Statology: <https://www.statology.org/assumptions-of-logistic-regression/>

Boston University School of Public Health. (2013, January 17). *Multiple Logistic Regression Analysis*. Retrieved from sphweb.bumc.bu.edu: <https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Multivariable/BS704_Multivariable8.html>

Brownlee, J. (2020, August 20). *A Gentle Introduction to the Rectified Linear Unit (ReLU)*. Retrieved from MachineLearningMastery.com: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>

chandan07. (2022, November 17). *Python for Data Science*. Retrieved from GeeksforGeeks: <https://www.geeksforgeeks.org/python-for-data-science/>

Chandramohan, P. (2020). *De-duplication Part 1: Deterministic Matching Approach*. Retrieved from MDMgeek: <https://www.mdmgeek.com/2014/05/21/de-duplication-part-1-deterministic-matching-approach/>

Clay, B. (2023). *Why Cancelling & Re-Booking is Killing Your Revenue*. Retrieved from IDeaS: <https://ideas.com/cancelling-re-booking-killing-revenue/>

dagnechaw, S. (2020, November 1). *What is Accuracy, Precision, and Recall? And Why are they Important.* Retrieved from Medium: <https://shiffdag.medium.com/what-is-accuracy-precision-and-recall-and-why-are-they-important-ebfcb5a10df2>

Heeswijk, W. v. (2022, January 31). *Precision and Recall - A Comprehensive Guide With Practical Examples*. Retrieved from Towards Data Science: <https://towardsdatascience.com/precision-and-recall-a-comprehensive-guide-with-practical-examples-71d614e3fc43>

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering, 9*(3), 90-95.

JMP Statistical Discovery LLC. (2023). *Multicollinearity | Introduction to Statistics*. Retrieved from JMP: <https://www.jmp.com/en_us/statistics-knowledge-portal/what-is-multiple-regression/multicollinearity.html>

Kurama, V., & Whitfield, B. (2022, August 31). *Feedforward Neural Networks: What is Feed Forward*. Retrieved from Built In: <https://builtin.com/data-science/feedforward-neural-network-intro>

Leung, K. (2021, October 4). *Assumptions of Logistic Regression, Clearly Explained*. Retrieved from Towards Data Science: <https://towardsdatascience.com/assumptions-of-logistic-regression-clearly-explained-44d85a22b290>

Lu, X. (n.d.). *Correcting the Quasi-complete Separation Issue in Logistic Regression Models.* Retrieved from SAS Support: <https://support.sas.com/resources/papers/proceedings16/10220-2016.pdf>

Nayak, R. (2022, April 26). *Focal Loss: A better alternative for Cross-Entropy*. Retrieved from Towards Data Science: <https://towardsdatascience.com/focal-loss-a-better-alternative-for-cross-entropy-1d073d92d075>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

prakharr0y. (2020, October 24). *Intuition of Adam Optimizer*. Retrieved from GeeksforGeeks: <https://www.geeksforgeeks.org/intuition-of-adam-optimizer/>

Putro, N. A., Septian, R., Widiastuti, W., Maulidah, M., & Pardede, H. (2021). *[PDF] PREDICTION OF HOTEL BOOKING CANCELLATION USING DEEP NEURAL NETWORK AND LOGISTIC REGRESSION ALGORITHM*. Retrieved from Semantic Scholar: [https://www.semanticscholar.org/paper/PREDICTION-OF-HOTEL-BOOKING-CANCELLATION-USING-DEEP-Putro-Septian/53670f41397ef61316c9e79337445afb26fb4882#related-papers](https://www.semanticscholar.org/paper/PREDICTION-OF-HOTEL-BOOKING-CANCELLATION-USING-DEEP-Putro-Septian/53670f41397ef61316c9e79337445afb26fb4882%23related-papers)

Raza, A. (2023). *Hotel Reservations Dataset.* Retrieved from Kaggle: <https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset>

Sanchez-Medina, A. J., & C.-Sancheq, E. (2020, August 1). *[PDF] Using machine learning and big data for efficient forecasting of hotel booking cancellations*. Retrieved from Semantic Scholar: <https://www.semanticscholar.org/paper/Using-machine-learning-and-big-data-for-efficient-S%C3%A1nchez-Medina-C.-S%C3%A1nchez/cbe04cd4096aa4edac703a9028e1551680cb183a>

The Investopedia Team. (2022, July 26). *Variance Inflation Factor (VIF)*. Retrieved from Investopedia: <https://www.investopedia.com/terms/v/variance-inflation-factor.asp>

The pandas development team. (2022, August 31). pandas-dev/pandas: Pandas 1.4.4. Zenodo. doi:[10.5281/zenodo.7037953](https://zenodo.org/record/7037953)

vartika02. (2021, July 29). *Disadvantages of Python*. Retrieved from GeeksforGeeks: <https://www.geeksforgeeks.org/disadvantages-of-python/>

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., . . . SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods, 17*, 261-272. doi:[10.1038/s41592-019-0686-2](https://www.nature.com/articles/s41592-019-0686-2)