

A Step-by-Step Approach

to

Analysing Data for Insights

Presented to:

Head of Data Science

ACC New Zealand

Analyst:

Abbas Khan

1. **Background:**

The current dataset was provided to be analysed for actionable insights related to segmenting and targeting ACC members to whom the app having information on safe gym practices should be deployed. The insights should focus on the features to be built into the priority, provide information on who among the current members are most likely to incur an injury during a workout, and hence be placed on a priority basis for deployment.

1. **Data Inspection:**

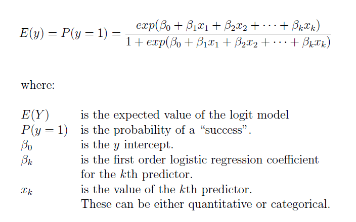
On inspecting the data along with the dictionary provided, there are 80,000 rows and 27 variables. The following points were observed:

1. Id of the ACC member to preserve personal information identification.
2. Numeric variables ranged from, age of the member, the number of total injuries while in the gym, injuries while weightlifting, type of injuries (shoulder, knee, hip, etc).
3. Categorical variables describing the location of the member, ethnicity, level of the strenuous nature of the member's work, the year of the last claim, and a variable “y” with values of yes (Y) and no (N).
4. An ordinal or ranked variable that included the socio-economic status in terms of deprivation index.
5. **Model Selection for Analysis:**

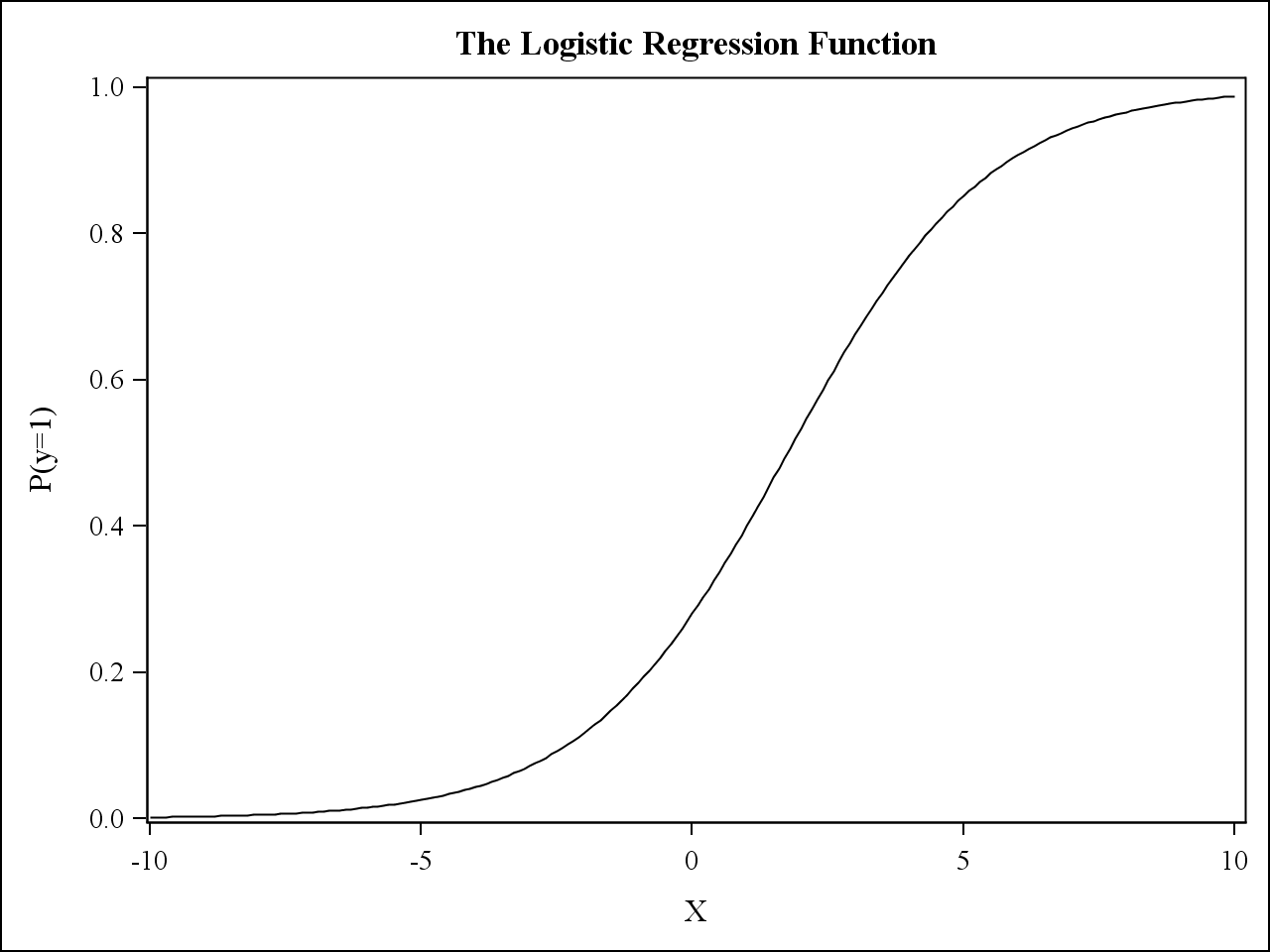
Given the nature of the variable “y” in the dataset that was “yes” if the member experienced an accident in the gym following the year of data collection and “no”, if he did not, made me decide to proceed with the logistic regression model.

Logistic regression or binary classifier model is used to predict the result of a categorical variable having two outcomes. As I am trying to “predict” the features to be included in the app as well as gauge the propensity of an injury among the members so that they can be targeted for app deployment on a priority basis. I am also interested in knowing the effect of the given variables on the incidence of an injury. The model works on the “maximum likelihood estimation” of the probability of an injury happening (dependent variable) given the selected and significant variable (independent variable) in the model.

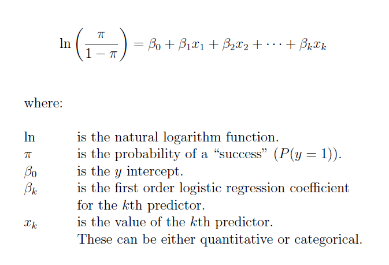
The equation of logistic regression is as follows:



The expected value E(Y) of the model will never fall below 0 or will go above 1. It’s a non-linear regression type model the result of which is a sigmoid curve.



The formula can then be converted onto a linear function by taking the natural log of the Odds Ratio, which is P(y=1) [Probability of Event Happening] / (1-P(y=1) [Probability of Event Not Happening]. This log of the Odds Ratio is termed as the logit and the new formula is as below.



1. **Out of Scope:**

Given the constraint of time, I decided not to proceed with Exploratory Data Analysis as the descriptive statistical methods provide the much-needed information to discover patterns, spot anomalies, test hypotheses, and check any assumptions using graphical representations. I hence started on created the predictive binary classifier model.

1. **Software Used for Analysis:**

I decided to use the R® software due to its availability as open-source software for analysis and my previous work experience on the same.

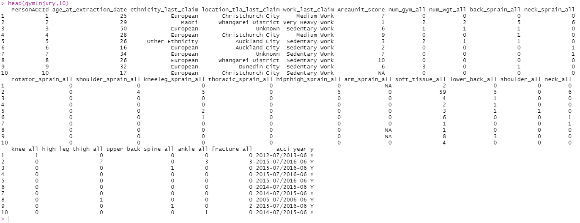
1. **A step-by-Step description of Methodology of Analysis:**

**Loading R packages, Data Sanity Checks, and Pre-Processing**

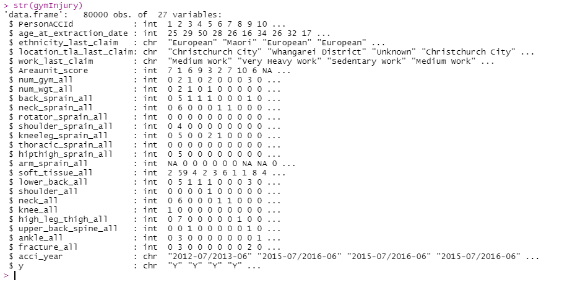
Loading the relevant packages in the R environment

Importing the dataset CSV into R followed by data inspection and sanity checks. These included

Checking the top 10 rows of each variable. Through this, I found that there are 27 key variables in the dataset.



Looking at the contents, class, and levels of each variable within the dataset. The data had to be reloaded with “Stringsasfactors = True” to analyse the levels of the categorical variables.



The next step was to find the frequency distribution of the “y” variable, which resulted in 4000 ACC\_IDs with an injury and 76000 without an injury (5% penetration).

Finding and treating observations or rows with missing (NA) values. I first found if null values existed within the dataset using the na.omit function in R. The result was a table with 46880 rows. Hence almost 41% of the dataset had a null value. These values were treated by replacing them with the median value of the corresponding variable.

The y variable having values of Y and N was replaced by a new variable “Y”, where the value Y was replaced by 1 and N by 0. This is required for creating the dependent variable for logistic regression.

Creating dummy variables for each level of a categorical variable. As the logit of odds ratio has a linear model, I needed to convert all categorical dependent variables into a dummy variable, with values 1 (if present for that member) and 0 (if not present). On creating the variables and looking at their names, I found certain variable names without an underscore. Hence added the same to those variables.

Certain variables were initially dropped. These included:

* + 1. The original variable y with values Y and N
    2. Accident\_year, which could have been proved useful in gauging the effect of recency of an injury on the model results, but was discarded due to its format and the time that would be taken to convert it into a date and calculating recency score
    3. Work\_last\_claim\_NA. This was created due to an NA value in the categorical table and which could not be replaced with any median value. This variable would not give any insights. Hence dropped.

**Splitting data into Train and Test Dataset**

The data will now be split into a train and test dataset in a 75:25 ratio. The model will be built on using the training dataset and the results would then be predicted using the test dataset. In usual cases, I generally have past data to train the model and then predict the outcome on the current on future timeline.

Setting Seed Number so that I have the same records for Test and Train data during each subsequent run

**Running the Logistic Regression Model and Variable Reduction**

The initial run using the glm() function did not give good results due to the following:

Certain variables had NA values and the rest all got significant p values.

The Training Data needs to be optimised through the "Stepwise" variable reduction process and then rerun for Logistic Output

I can also check for Multi Collinearity between the dependent variables, but these would partially be taken care by the "Stepwise" function

Also, certain variables need to be removed before the "Stepwise" procedure such as

1. Variables with Estimates as NA ('location\_tla\_last\_claim\_Whangarei\_District' & 'ethnicity\_last\_claim\_Residual\_Categories')
2. Variables representing Member ID ('PersonACCId')
3. Variables from which no insights can be derived ('location\_tla\_last\_claim\_Unknown')

Post running the Stepwise function and rechecking the model, I could find certain insignificant variables and decided to rebuild the model with the following ones:

age\_at\_extraction\_date

ethnicity\_last\_claim\_European

location\_tla\_last\_claim\_Auckland\_City

location\_tla\_last\_claim\_North\_Shore\_City

location\_tla\_last\_claim\_Wellington\_City

work\_last\_claim\_Heavy\_Work

work\_last\_claim\_Light\_Work

work\_last\_claim\_Medium\_Work

work\_last\_claim\_Sedentary\_Work

Areaunit\_score

num\_gym\_all

num\_wgt\_all

back\_sprain\_all

kneeleg\_sprain\_all

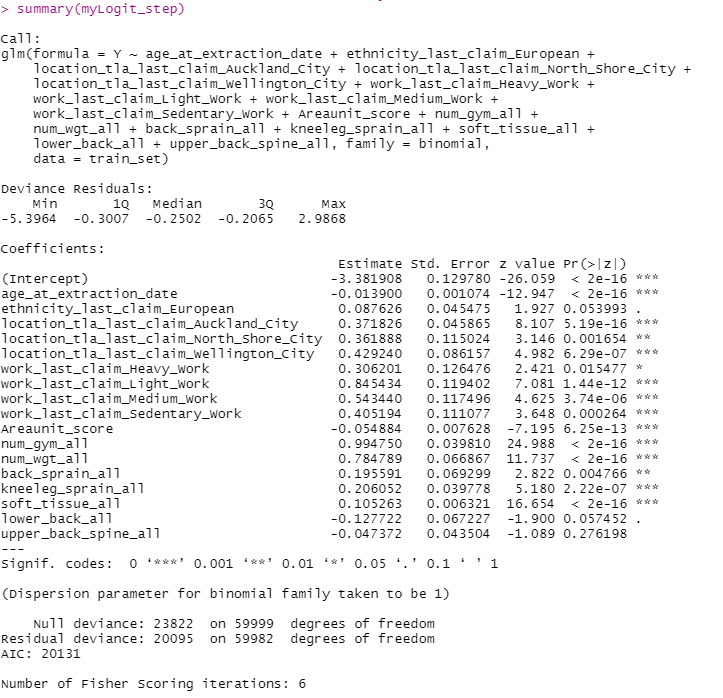
soft\_tissue\_all

lower\_back\_all

upper\_back\_spine\_all

The final summary result is as below

:



Here I can see that I have picked the most significant variables. Where p values are mostly at 0.001 (less than 0.05 for 95% statistical confidence)

**Preparing Summary Table**

I now calculated the odds ratio for each of the corresponding variables which will signify the association between the dependant(y) and the independent variable(x). It provides the outcome that the injury will occur given the presence of that variable compared to the odds of that variable not being present.

I then proceeded to standardise the coefficient estimates (beta estimates) of each selected variable within the model which are used to provide a uniform scale of measurement for the existing coefficients and to measure the weight/contribution % of a particular variable on the overall model.

This final table was exported and is as below:



(Present in Excel – gymInjury\_Model\_Result)

**Predicting the results from train dataset to test data and model diagnostics**

The results of the model created from the train data now have to be validated on the test dataset.

This was by creating and assigning the predicted probability of an injury occurring to the members in the test dataset.

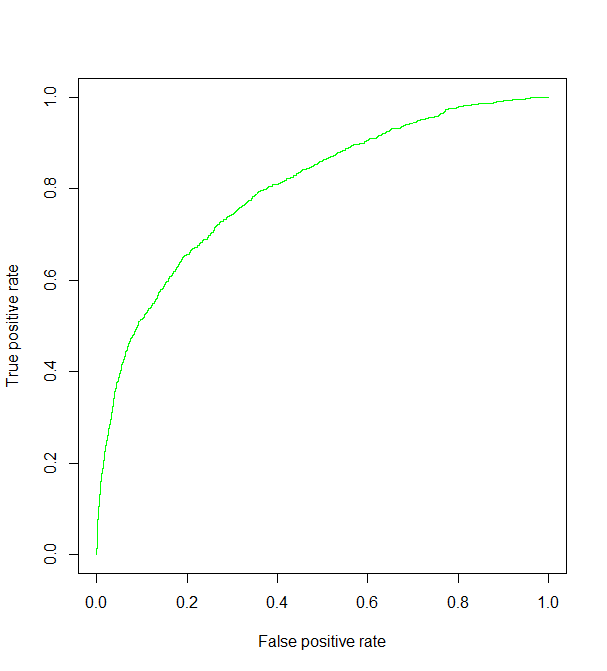
The model needs to be now diagnosed against various metrics so that it can be finalised and validated for future implementation. These include:

**Accuracy**: is derived from the Confusion Matrix and defined as (True Positives + True Negatives) / (All Positives + All Negatives). For the test set, this is calculated as 0.9515000 being the max value, which makes the model highly accurate

**Cut off Value:** which defines the data values being treated as positive/negative was calculated as 0.6811167

**ROC Curve and AUC:** Stands for Receiver Operating Characteristic, the curve plots the True Positive Rate versus the False Positive Rate for all possible cut-off values. The Area Under the ROC Curve (AUC) measures the usefulness of the model and the larger the area, the more useful the model is. The AUC of our model is 0.8080222 and projects a good model diagnostic

**ROC Curve**



***TPR = TP / (TP + FN)***

***FPR = FP / (FP + TN***

**Lift Chart:** This shows the actual lift in the success of implementing the model. The lift value is the ratio between the result predicted by the model and the result using no model. The chart predicts a high lift of close to 20% of the positive predictions.



**KS Statistics:** is the maximum difference between cumulative true positive and cumulative true and is used as the metric to judge the efficacy of the model. A KS value is max (high) at 1.0 and min (low) at 0. The KS score derived here is0.4607368 which is close to 0.5 and is a good score.

**Confusion Matrix:** The confusion matrix is 2x2 matrix that Actual (True) Positives and Actual(True) Negatives against Predicted Positives and Predicted Negatives. The confusion matrix of our model is as below.

y\_pred

0 1

0 18922 (TRUE NEGATIVE) 78 (FALSE POSITIVE)

1 903 (FLASE NEGATIVE) 97 (TRUE POSITIVE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| y | Y pred | **Recall**  Out of all Positive classes, how much I predicted correctly | **Precision**  Out of all Positive classes predicted correctly, how many are positive | **Accuracy**  Out of all classes, how much I predicted correctly |
| 0 | 0 | TP / (TP + FN) =  0.097 | TP / (TP + FP) =  0.55 | (TP + TN)/(TP + FP + TN + FN) =  0.95 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

Key:

P (\# positive samples), N (\# negative samples), TP (\# true positives), TN (\# true negatives), FP (\# false positives), FN (\# false negatives).

The last part of the process post model evaluation was to combine the test and train dataset, assign the predicted probability, group the members into deciles, and take the top 1000 members who had an injury in the past with descending probability of injury occurring from the 1st and 2nd decile groups.

I am attaching the R Code along with the Test, which has all the necessary comments for running the model at your end.

For Further Information Contact

Abbas Khan || [Abbas.Khan@Outlook.in](mailto:Abbas.Khan@Outlook.in)