# Task 2 submission: Design

This is your submission document for [DataCapX Submission 2: Design](https://courses.edx.org/courses/course-v1:AdelaideX+DataCapX+3T2018/courseware/a7d3a888e3aa4b2689c5421bc4550619/beb0b3fe73c4426f9343e232ec83375b/?activate_block_id=block-v1%3AAdelaideX%2BDataCapX%2B3T2018%2Btype%40sequential%2Bblock%40beb0b3fe73c4426f9343e232ec83375b).   
Save this document on your local machine and include all of your work within the relevant part of the assignment. Once you’ve completed every part of Task 2, upload this document via the [Your Response area](https://courses.edx.org/courses/course-v1:AdelaideX+DataCapX+3T2018/courseware/a7d3a888e3aa4b2689c5421bc4550619/beb0b3fe73c4426f9343e232ec83375b/?activate_block_id=block-v1%3AAdelaideX%2BDataCapX%2B3T2018%2Btype%40sequential%2Bblock%40beb0b3fe73c4426f9343e232ec83375b).

**Note: If there are scripts related to this task, make sure you submit them when you submit this document.**

# Checklist

* Have you answered every question?
* Have you shown all of your working, including evidence of your code?
* Have you clearly stated conclusions where required?
* Have you saved your code in a script (if applicable)?

# Task 2 submission: Design

1. Articulate which classification methods could be used for this task. Articulate the way to compare the classification models.   
   [2 points]

The requirement is to predict the brand\_safe factor which has TRUE/FALSE values in the dataset. We therefore need to build a **binary classification model**.

The following are some of the possible ways to create this classification model:

* **Generalized Linear Model** – This is one of the easiest ways to solve this problem. It has the advantage that it readily provides information criteria such as AIC and BIC that can be used for comparing models. In addition to this it also provides an easy method of applying stepwise regression to select the optimal variables to use in the model. In relation to “explainable AI” the model produced with this method can more easily be interpreted and is less of a black-box.
* **Apache Spark** –Using the sparklyr library suitable models can be created in Spark. This is advantageous when working with big data to allow for distributed processing which could not otherwise be analysed or modelled using a single PC/server. The following machine learning algorithms would be particularly suitable to this problem:
  + ml\_random\_forest
  + ml\_decision\_tree
  + ml\_gradient\_boosted\_trees
  + ml\_multilayer\_perceptron
  + ml\_naive\_bayes
* **H2O** – Something like H2O is usually my first choice for doing a proof of concept before investing too much time into data wrangling, cleaning, feature extraction and modelling. It provided algorithms similar to Apache Spark, but also goes beyond that with some more algorithms and methods. I would normally just throw a random sample of my data at the [AutoML](https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html#code-examples) feature to get a rough idea of what lies ahead.

As a preliminary I have compared GLM with a few of the Spark algorithms using AUC as the evaluation metric:

* glm : 0.7023
* ml\_logistic\_regression : 0.7009
* ml\_random\_forest : 0.7007
* ml\_gradient\_boosted\_trees : 0.7010

Based on these initial results I believe it would be sufficient to simply create a generalized linear model and focus on optimizing the features used.

When comparing models there are a few common metrics that works well in practice, the two most relevant in this scenario is:

* **Area under the ROC Curve** – as used above to help me decide what try first, as well as the accompanying receiver operating characteristic curve plot as taught in the course.
* **Akaike information criterion** or **Bayesian information criterion** – AIC and BIC is great means to compare models and aid with model selection. It helps us to compare the balance between model complexity and ability to explain the data.

*I might have complete miss interpreted what this question was really asking but it is perhaps worth mentioning, as a personal insight, depending on the libraries I use, instead solving this problem with a binary classifier I sometimes treat it as a multi-class problem, as that is sometimes the only way to get the confidence scores.*

1. Articulate which criterion could be used for evaluating the model's ability to predict new data. Specify how this criterion is calculated.   
   [2 points]

Once a model is selected, possibly using the information criteria mentioned in the previous section, the model should be evaluated to measure how well the model “fits” the data and how much skill it has in predicting to correct class. The following are two of the key factors to look at:

* **Confusion Matrix** – The confusion matrix is a great tool, especially when communicating to business how a model is performing. When compiling the confusion matrix the True Positives, False Positives, True Negatives, and False Negatives values needs to calculate. To obtain these values is fairly simple and all one needs to do is compare the true values with the predicted values. From these one can then calculate the following single number metrics which I find the most useful when comparing models:
  + ***Precision*** - What proportion of positive identifications was actually correct?

TP / (TP + FP)

* + ***Recall*** - What proportion of actual positives was identified correctly?

TP / (TP + FN)

* + ***The F1 score*** – Defined as the harmonic mean of the precision and recall. Or more simply the balance between the two metrics above.

[2 \* (precision \* recall)] / (precision + recall)

Finally I also like to show the **Gini Coefficient** (2 \* AUC – 1), but I have learned to pick wisely when presenting models to less numerically minded colleagues. It is often best to just go with things that are the easiest to explain and understand. The ROC plot is also always a hit with the crowds…

* **Cross-validation** – It is always important to separate your data into a training set and a test set that was not seen during model training. The above metrics can then be calculated and compared with the metrics from the trading set. When there is too large a difference between the two it can be an indication of either not generalizing, or overfitting. If overfitting is suspected this can be taken further and a k-fold cross-validation can be done for verification.

Articulate how you would select the input factors based on the criterion. Describe two methods to achieve this.   
[5 points]

If I go with my decision to build a Generalized Linear Model selection of the best input factors to use will definitely be best handled by Stepwise Regression. In short the idea of this method is, for example in backwards elimination, to remove the worst performing factor, rebuild the model, go to the next step and keep repeating until we find the optimal selection.

There is enough combinations using this method to give us a variety of different ways to get to the optimal factor set.

First we can chose if we want to look at AIC or BIC values to see if a model is better or worse. Then we can chose to do *forward selection* (starting with an empty model and add factors)*, backwards elimination* (starting with all the candidate predictor values and start removing), or even burning the candle at both ends with bidirectional elimination.

For the purposes of this piece of work I will chose the following:

* Akaike information criterion (AIC) as the
* Perform forward selection starting with a null model.
* Perform backward elimination from the full model.

The final AIC of the two methods will then be compared to determine which achieved the best results. If there is no distinguishable difference with bidirectional elimination can also be tested.

*As an alternative we could perhaps also simply look at which coefficients looks the least significant in the model and experiment manually adding and removing factors.*

Total points possible for Task 2: Design 9