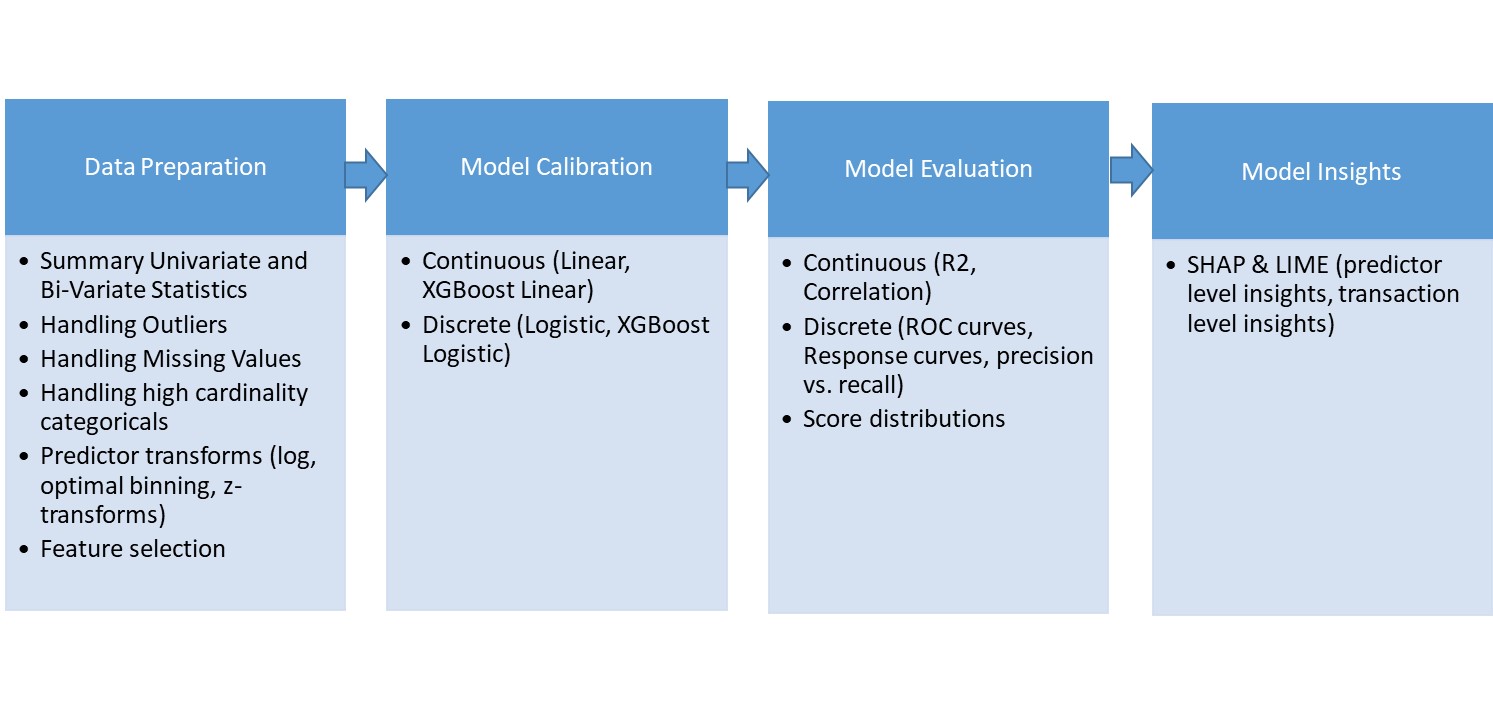
# NEMESIS-PREDICT – High level specifications document

The goal of the project is to set up an end-to-end pipeline for auto-calibrating a predictive model. It will start with preprocessing the data to create predictors, calibrate a model, and produce performance charts, and provide predictor insights.

## Product Description

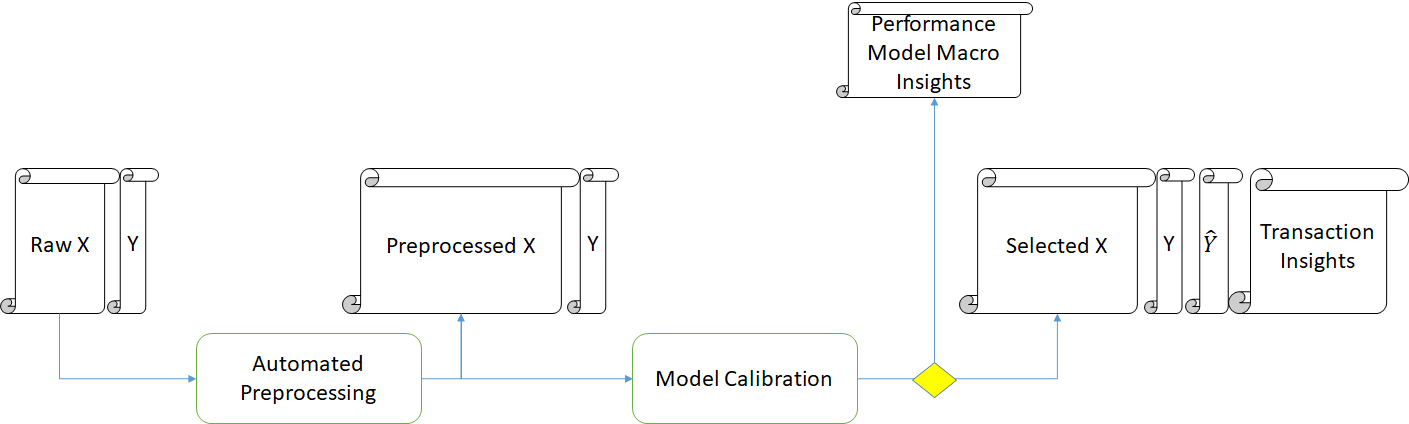
At a very high level, the pipeline can be captured by the following diagram. Given a dataset, with a target (what is to be predicted) and a list of potential predictors, the pipeline is intended to

* Clean and preprocess the data – this includes handling common issues with raw data such as presence of large outliers, dealing with missing data, do some preliminary screening, gather univariate and bivariate stats, and then come up with transformed predictors (such as log-transformed). For some of these tasks, open source packages are available (for example, profiling packages are available that does the uni-variate and bi-variate stats, there are packages available for optimal binning). This task can be guided by the automated data preparation algorithms available in the attached algorithms guide.
* Calibrate a predictive model – based upon the target and the preprocessed predictors resulting from the first task above, this task will either calibrate a continuous or a discrete predictive model. The requirement is to have the capability for one simple (linear for continuous, and logistic for discrete) and one advanced method (XG-Boost Linear for continuous, and XG-Boost Logistic for discrete). Open source packages should be researched and adapted for this capability.
* Produce evaluation results for the generated model – based upon the calibrated model, produce the standard performance evaluation visuals and statistics. The goal is to produce these graphics and statistics in a way that is agnostic of the model. This module should use only two columns – predicted & actual, and it should be able to produce all the statistics. Additional summary charts could include how the model performance improves as more predictors are added (for example).
* Derive insights into the guts of the model (model explainability)– There are two levels of insights that can be gleaned from any predictive model. Macro-level insights inform how strong a predictor is, and how it is related to the target (for example, age (x variable) is positively correlated with income (y variable). Transactional level insights refer to the contributions made the predictors for a particular score (why did this claim get a high fraud score, what factors contributed to the high score?). Here, the open source packages SHAP (inspired by Shapely Regression) and LIME (Second option) should be leveraged to derive both the types of inputs.



## Data flow Design

Note: Scroll bars are all panda data-frames.





## Datasets

### Discrete

Anonymized Audit Dataset

### Continuous

<https://www.kaggle.com/c/ClaimPredictionChallenge/overview>