Home Loan Credit – Statistical Inference

Introduction

This analysis includes implementation of new variables for significant missingness regarding the target variable, missing value imputation, encoding for categorical variables, normalisation, feature selection and logistic regression using the loan application data [1]. We learned that there are significant missing values, and they seem to affect the outcome of the target variable from the earlier study [2]. So we here implement indicator variables for those in advance to any further steps.

Our target variable is the indicator of risk not to repay their loan, and we want to build a logistic regression model to predict the target correctly from a large number of predictor variables. Encoding is a way to change categorical values to numeric to make modelling easier - we apply label encoding for variables with less than or equal to two categories and one-hot encoding for ones with more than two categories [3]. Missing value imputation will be done with the median of observed data in each variable and followed by normalisation to eliminate the effect of noises and ensure the model is reliable.

In this analysis, we want to go a bit deeper in feature selection engineering - applying three different techniques - namely feature importance, recursive feature elimination and univariate selection - and compare their ROC AUC metric scores [4].

Missing indicator variables

We recap that there are 307511 observations in total – only 8.1% of data have the target outcome of 1, which expected to have difficulties to repay their loans. Over half of the features have some missing values, in number, there are missing data in 64 features among the total of 122 –with the maximum of 70% of missingness. From Chi-square test we found that missingness is significantly related with the target outcomes in 62 features; so we make 62 indicator variables for these – the value of one implies missing. We have the dimension of (307511, 184) after this exercise for out data. After encoding step, we impute the missing values using their median.

Encoding and normalisation

Logistic Regression modelling requires all of the feature values in numeric, so we need to transform the string value to numeric –to maximize our information level during this change, we use label encoding for two-level category variables and one-hot encoding for three- or more-level category variable. We have the dimension of (307511, 305) after this exercise for out data.

Sometimes the range of values for a feature are big, and they lead unreliable model, i.e. they give a good score for the given data, but really bad predictions for a new data. To avoid this unacceptable result, we normalise all features in a range of (0, 1) using MinMaxScaler.

Feature selection

Feature selection is the main part of this analysis to investigate any differences in ROC AUC scores from the following three different methods under the same circumstances:

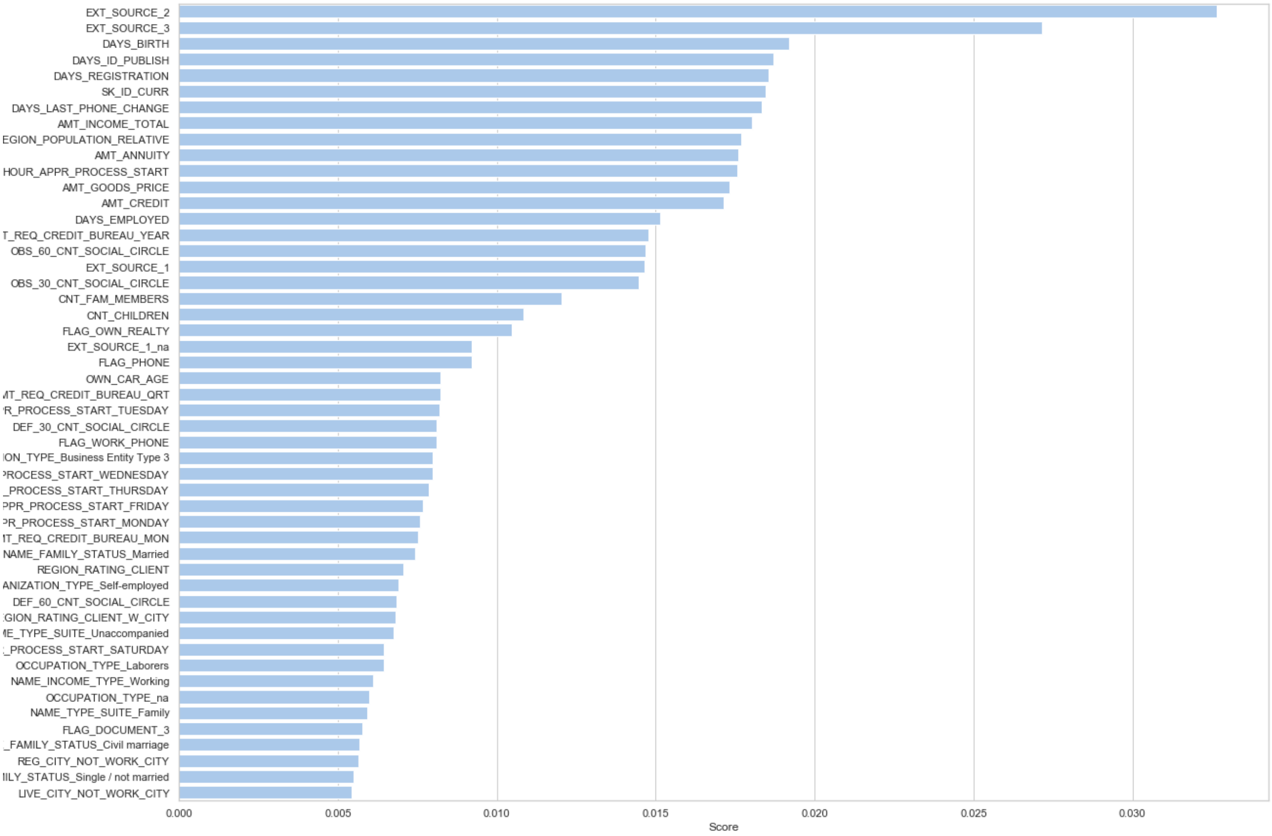
- Feature importance
- Recursive feature elimination

- Univariate selection.

Feature importance

We apply ExtraTreesClassifier with default parameters and get the sorted score list from the highest to lowest. The highest score is 0.033 on EXT_SOURCE_2 and followed by 0.027 on EXT_SOURCE_3 as shown in Figure 1; we also can see some of the missing indicator variables are in the list of the top 50 important features. We decide to choose all features that are scored over 0.002, and the total number of 136 features are selected to predict the target value in our logistic regression model. The ROC AUC score for the model is 0.7488.

Figure 1: Top 50 significant features



Recursive feature elimination

RFE is computationally expensive; it takes a long time compared the other algorithms because it computes scores all the time from the full model until getting the model with the number of features that we provide in advance using the elimination of one feature each time. The features selected from this method are different from the previous ones, but score increased a fraction to 0.7494.

Univariate selection

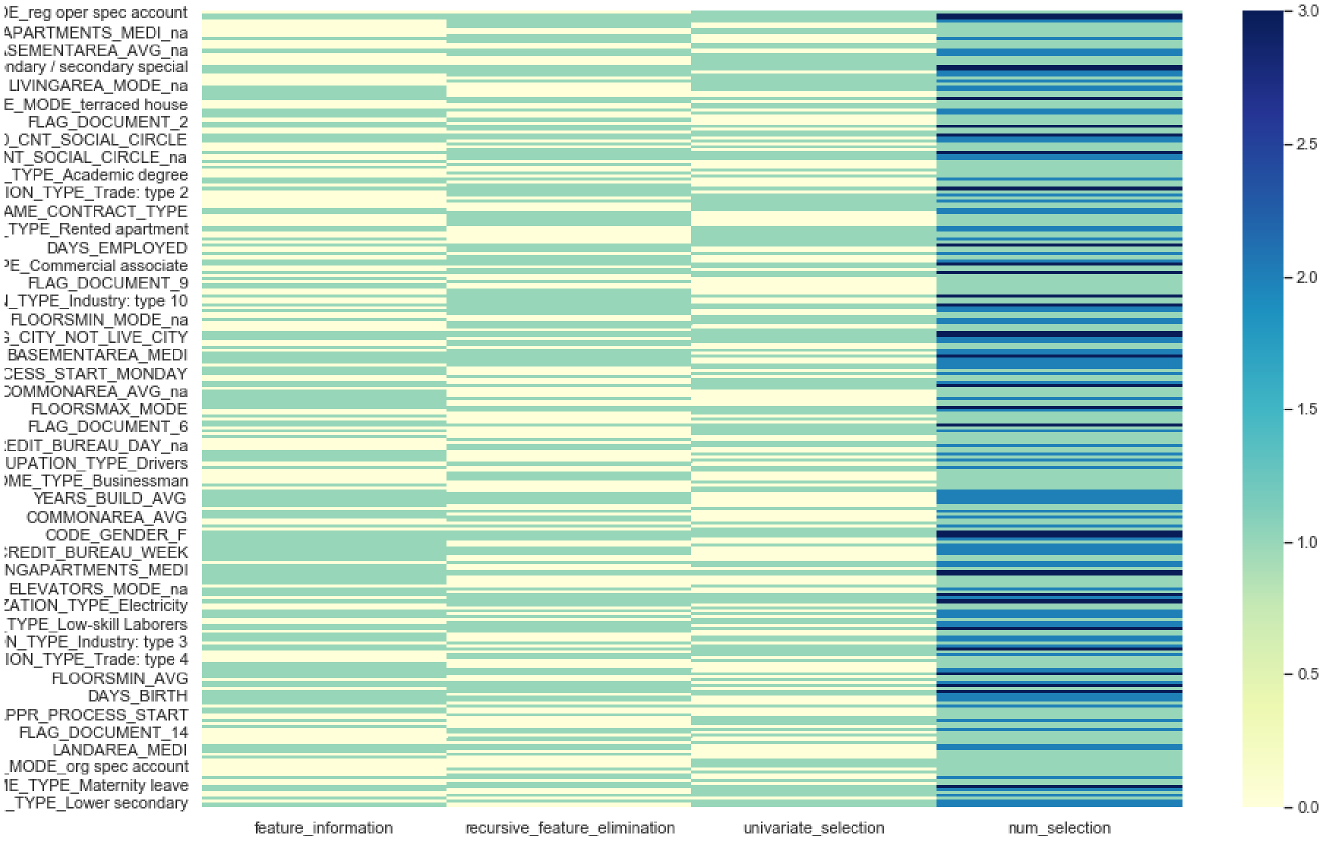
Opposite to the RFE, univariate selection uses a single variable to compute test score with the target variable – we use Chi-square test in this process, and computational demand is relatively light. This algorithm has a score of 0.7442, the lowest among these three algorithms we try.

Comparison features

We recall that the number of features for each model is 139 – since our three algorithms take different sets of features, we combine all three sets and compute the number of inclusion. There are 33 features that are included in all three models and 87 and 147 included in two and one model respectively among the total of

266 features combined from the three methods, as shown in the right-quarter of Figure 2. Left three quarters of the figure represent the selection of features for each algorithm, e.g. feature importance, RFE and univariate selection method from the very left.

Figure 2: Feature selection outcomes of three different algorithms



What is next?

This analysis is done only for the application dataset and does not include other historical data – so after combining those data in aggregation, the model outcome could change significantly. Any algorithm has hyper-parameters to find the selection of features under the given criteria; here we just use the default setting for these, but we might need to change the settings to optimise the outcome. There are no validation and test processes in this analysis, and we used the whole set of data – to generalise our model we want to validate and test the model; this requires splitting dataset in advance.

References

[1] https://www.kaggle.com/c/home-credit-default-risk/data

[2] https://github.com/DianePark/Predictive-modelling-with-Python/blob/master/03_HLC_Wrangling.ipynb

[3] https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/

[4] https://machinelearningmastery.com/feature-selection-machine-learning-python/