# Credit Scoring: Feature selection on machine learning algorithms

Centre for
Data Analytics
Insight

David Browne, Dr. Steve Prestwich

## Objective

To introduce and refine algorithms to detect good/bad loan customers. The machine learning algorithms used were Logistic Regression (LR), Naïve Bayes (NB), Random Forests (RF) and Support vector Machines (SVM).

## Introduction:

The 3 credit datasets used were an Australian, German and Japanese credit datasets from the UCI machine learning repository [1], and are described below.

| Dataset | No. of instances | No. of numerical features | No. of ordinal features | No. of nominal features | Class 1 : Class 2 |
|---|---|---|---|---|---|
| German | 1000 | 3 | 4 | 13 | 700:300 |
| Australian | 690 | 6 | 0 | 8 | 383:307 |
| Japanese | 684 | 6 | 0 | 9 | 303:381 |

All 3 datasets had 2 classes, thus making it a binary classification problem. Class 1 was a good credit risk while class 2 was a bad credit risk, and from the table it can be seen that the Australian and Japanese datasets were reasonable balanced while the German dataset was imbalance hence make it more challenging to predict the class.

## Data Preparation

Discretization is the process of dividing continuous features into groups (bins) and for this project varies types of supervised and unsupervised methods were analysed, including equal frequency intervals and Chi-squared correlation. It can be seen that by using Chi-squared binning on the Australian dataset had nearly a 7% increase in accuracy when the Naïve Bayes algorithm was applied.

| Discretization | Accuracy | F-Score | Sensitivity | Specificity | Precision |
|---|---|---|---|---|---|
| None | 0.787 | 0.8245 | 0.904 | 0.6421 | 0.7578 |
| Chi Squared | 0.8555 | 0.8722 | 0.8905 | 0.8122 | 0.8545 |

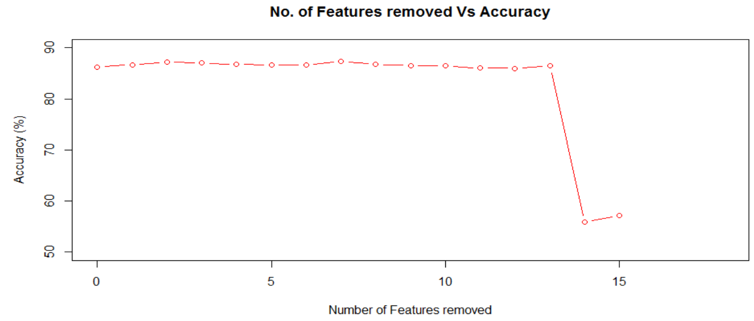
## Feature Selection

There are three main methods for feature selection, the filter method, the wrapper method and the embedded method. The first two will be analysed in this project on the three credit datasets using the four algorithms mentioned above. To validate the results of the models, Monte Carlo 100-fold cross-validation was used, which helps to limit the problem of overfitting. The three datasets were partitioned randomly 70% for training and 30% for testing during each fold of the cross-validation.

### Filter Method

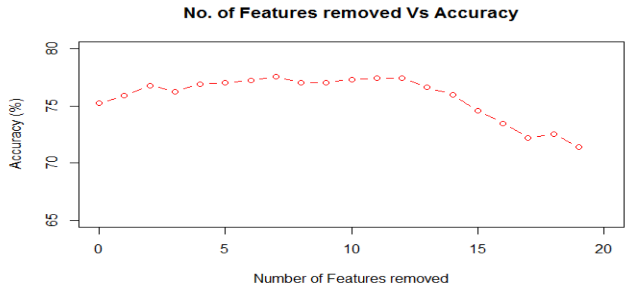
Each feature is ranked according to their correlation to the target variable using statistical metrics, such as Entropy, Goodman Kruskal Lambda and Messenger Mandell Theta. Backward elimination is applied to the full set of features removing one feature at a time starting with the lowest ranking feature, which is the one that is the least correlated to the target variable. The following graph is an example of the Naïve Bayes algorithm results from the Japanese dataset using k-means discretization and Entropy as the statistical correlation metric. By removing 13 features the overall accuracy of the algorithm loss none of its predicting power, in fact, there was a 0.25% increase in accuracy compared to using all the features.

There were a couple of peaks in the graph where there was a 0.5% increase, 1, 2 & 7 features removed, but since this is an algorithm to reduce the number of features, removing 13 would be the optimal option.



### Wrapper Method

A learning algorithm, such as Random Forest, Naïve Bayes or Support Vector Machines in this project, is used to search and evaluate optimal feature subsets. Depending on the size of the search problem, a random hill-climbing , simulated annealing or greedy backward elimination may be applied. In the latter method, at each iteration, the set of features is analysed and an evaluation metric is used to determine the worse feature which is then removed.



The graph above shows the greedy backward elimination wrapper method preformed on the German dataset using equal-width bin discretization and Random Forest as the learning algorithm. It can be seen that removing 12 features yields the optimal solution for the feature selection, increasing the overall accuracy by 1.6% compared to using the full set of features. Below is a table comparing this projects work, in red, to current results, in black, from a paper released this year [2]., showing it out-performs the current results.

| Dataset | No. of Features removed | Accuracy | Algorithm |
|---|---|---|---|
| German | 12 | 77.42% | Random Forest |
| German | 12 | 67.10% | Random Forest |
| German | 14 | 73.65% | Naïve Bayes |
| German | 12 | 70.80% | Naïve Bayes |
| Australian | 10 | 86.67% | Random Forest |
| Australian | 2 | 84.92% | Random Forest |
| Australian | 12 | 85.53% | Naïve Bayes |
| Australian | 2 | 81.30% | Naïve Bayes |

## Conclusions and Future Work

- Feature Selection reduces overfitting, improves accuracy and reduces training time
- Develop pre-feature selection algorithm to remove redundant and irrelevant features

## References

[1] UCI repository of machine learning databases. Department of Information and Computer Science. http//www.ics.uci.edu/~mlearn/MLRepository.html.

[2] Wang, F & Liang, J (2016). An efficient feature selection algorithm for hybrid data.

This project has been funded by Science Foundation Ireland grant SFI/12/RC/2289