Module 1 assignment:

AI solution implementation

# Introduction

This report outlines the proposed implementation of an AI-based solution for a small fintech consulting start-up. Their business model is to identify potentially successful start-ups, and then help them build their financial models and progress through successful fundraising rounds.

The problem to be addressed is the prediction of start-up growth and successful future acquisition to improve targeting of suitable customers (Dellermann et al., 2021; Potanin et al., 2023). By improving customer acquisition and resource allocation, such an approach could greatly increase return-on-investment for the company and allow it to provide its services at lower costs.

# Development framework

Development of the proposed solution followed the Cross-Industry Standard Process for Data-Mining (CRISP-DM) framework, the *de facto* standard for developing and deploying data-mining projects (Schröer et al., 2021). The standard includes 6 stages, addressed as outlined below:

1. Business understanding – the company needs an accurate method to predict start-up success, as this will determine whether the time invested in helping them grow will lead to future returns. The model itself can be also used as a selling point to attract customers (Buchanan, 2019), potentially providing a competitive advantage on many levels.
2. Data understanding – the next stage involves identifying and exploring a suitable dataset containing data relevant to the business case and modelling problem. This includes developing an understanding of each attribute (i.e. variable or column) in the data and of relationships between them, and general descriptive analyses. Ideally, the dataset would be collated for purpose and using proprietary data (Bessen et al., 2022), but for this illustration a publicly-available dataset will be used.
3. Data preparation – after exploratory data analysis, relevant variables will be selected to train the prediction model, along with any transformations required to extract useful information from the data available (a process known as feature engineering; Duboue, 2020)
4. Modelling – using the resulting dataset, a number of different candidate models will be developed and fine-tuned in order to identify the optimal underlying parameters to be used each model (i.e. hyperparameter tuning; Russel and Norvig, 2021).
5. Evaluation – the performance of the resulting models will then be described and compared using a range of metrics; this includes developing an understanding of errors made by the model, and if possible of how predictions are made.
6. Deployment – the development process terminates with deployment of the model within existing workflows. This requires a thoughtful implementation and maintenance strategies, including building streamlined user interfaces, integration of real-time data feeds, user training and support, and monitoring of model performance.

For the purpose of this report, the free software WEKA (Waikato Environment for Knowledge Analysis, v3.8.6) will be used for a preliminary demonstration of the possibilities of the proposed solution (Frank et al., 2016). WEKA allows the development of machine learning (ML) models with a point-and-click approach (running on a Java backend), which facilitates early development and exploration (Bell, 2020).

# Data source description, exploratory data analysis, and feature engineering

A publicly-, freely-available Kaggle dataset was used for this use-case. The dataset (“Startup Success Prediction”) contains information on 923 real-world US-based start-ups, and can be accessed at https://www.kaggle.com/datasets/manishkc06/startup-success-prediction/data.

This dataset was chosen as it includes information on a reasonably large number of companies, including 49 attributes relating to location, sector, and previous funding rounds and milestones, as well as a variable for “status” (acquired or closed) which can be readily used as target class for prediction (Figure S1).

The data included 4 identifier attributes, 12 attributes associated with geographical location, 12 associated with business sector, 18 for fundraising and milestones, and 3 for outcome (i.e. acquired or closed) (Table 1). Companies included were located across 221 distinct US cities and 35 states, founded between 1984 and 2013, and with funding rounds and outcomes collected from 2001 to 2013 (Table 1).

After exploratory data analysis, 32 attributes were identified as potentially relevant and selected for modelling, after excluding of repetitive or highly-correlated attributes (based on their description, e.g. coordinates and city) and focusing on aggregated attributes where relevant (e.g. focusing on state-level location, and with one-hot encoding for particular states; Figure S2). Of note, many of the attributes included in the raw data were duplicates from one another (e.g. several copies of company identifier, or different expressions of geographical location), one-hot encoded attributes for particular US states or business sectors, or computed attributes (e.g. age of company at first successful funding round or milestone, based on founding and first funding dates). The selected attributes were further investigated visually to exclude the possibility of co-linearity (data not shown). Data transformations were applied as required in order to convert attributes to the correct data type (e.g. numeric to nominal for outcome labels), as outlined in Table 1. Distributions were assessed visually for numerical attributes (Figures S3-S10), after which an additional mathematical transformation was applied to log-transform total amount of funding received (Figure S10). The dataset was provided without any missing values, i.e. all cells have a value except for cases where one should not exist (e.g. date of closure for companies that did not close).

Target class imbalance was found, with most companies observed to achieve success (i.e. acquisition; 597/923 or 64.7%). Class rebalancing was undertaken using Synthetic Minority Oversampling Technique (SMOTE), which yielded a new total number of instances of 1249 (652 fails, 597 successes).

# Problem conceptualisation and identification of candidate algorithms

The solution under development represents a classification problem (i.e. predicting a binary label) for a known label (the success label), and can therefore be tackled using supervised ML approaches. While other ML paradigms could be considered (e.g. unsupervised ML with clustering to identify companies which stand out from others in some dimension), the availability of a well-coded target variable of interest for all observations renders supervised ML a more useful and interpretable framework.

Myriad candidate algorithms could be selected for assessment. For this proof-of-concept, commonly-used and well-established algorithms were chosen to keep the proposed solution streamlined and focused on the potential capabilities of ML in this scenario, rather than the technical or computational aspects required for an industry-grade solution. The algorithms chosen were:

1) logistic regression (LR);

2) naïve Bayes (NB);

3) decision tree (DT);

4) random forests (RF);

5) k-nearest neighbours (KNN); and

6) support vector machines (SVM).

All of these are standard supervised ML algorithms, representing a range of fundamentally distinct mathematical classification approaches, therefore multiplying the likelihood of identifying an algorithm with adequate performance. The models selected also include highly-explainable and relatively simples modes (e.g. LR, DT, NB) and more complex ones (RF, KNN, SVM).

# Performance evaluation

Model performance was evaluated with three aims: 1) identify the optimal hyperparameter settings for each model, 2) compare model performance across different models, 3) estimate the generalisation accuracy of the selected model on unseen data (Raschka, 2018).

Model performance was assessed via 10-fold cross-validation (Russel and Norvig, 2021). Alternative methods such as holdout and bootstrapping could also be used. However, cross validation is better suited for small-to-medium datasets than holdout, and is less computationally intensive than bootstrapping (Raschka, 2018; Russel and Norvig, 2021). A fold number of 10 was chosen as this is considered the standard approach when using cross-validation, and is supported by empirical evidence (Raschka, 2018; Russel and Norvig, 2021).

For the purpose of hyperparameter tuning, model performance was estimated before class rebalancing. Once optimal hyperparameter settings were identified for each model, the impact of class imbalance was estimated by comparing model performance before and after class rebalancing.

Multiple metrics may be used to assess model performance. In the interest of simplicity, the following commonly-used metrics were chosen: accuracy, recall, precision, F1-score and area-under-the-Receiver-Operator-Curve (AUROC). However, other metrics have been suggested (Foody, 2023). For hyperparameter tuning, accuracy, F1-score, and AUROC were used as they provide more holistic measures of model performance (vs precision/recall), with accuracy prioritised for hyperparameter selection as a more simple and interpretable metric (F1-sore and AUROC used as complements).

Tables 2-5 show the hyperparameter tuning steps applied to each model (excluding logistic regression and naïve Bayes, as hyperparameter tuning does not apply to these). For DTs, model accuracy was improved from a baseline of 72.3% to 76.1% by lowering the confidence factor for tree pruning from 0.25 to 0.05, and increasing the minimum number of objects per leaf from 2 to 10. For RFs, none of the alternative settings improved default model accuracy (79.3%). For KNN, model accuracy improved from 61.8% to 67.5% by increasing the number of neighbours from 1 to either 7 or 10, with the later yielding higher F1-score and AUROC. For SVM, model accuracy was improved from 74.5% to 74.9% by allowing construction of calibration models and increasing the complexity parameter from 1 to 2. Of note, prioritising a different performance metric could lead to selection of different hyperparameter settings, but leading to small differences in performance.

Table 7 depicts a comparative assessment of model performance across all candidate models using the best performing hypeparameter setting in each, both before and after class rebalancing. Figures S11-S21 show WEKA screenshots for the underlying performance evaluation and attribute weighting/selection outputs. Baseline accuracy for this assessment was 64.7% and 52.2% before and after class rebalancing. RFs were the most accurate model in both settings (79.3% and 82.1% accuracy, respectively), outperforming other models across all metrics. Of note, precision, recall, and F1-score for the target class (1, i.e. success) were higher before class rebalancing. The next best performing algorithms were SVMs (74.9% and 79.7% accuracy) and LR (74.0% and 79.6%). Analysis of RF attribute importance suggests the most informative attributes were those related to total amount of funding, number of milestones and relationships established, and company age at different milestones, with location and sector having less importance (Figure S18). Interestingly, relationship number was also the attribute used in the first DT node (Figure S16) and the one with the highest weight in SVM (Figure S21), but the relative importance of each attribute varied widely across models. Of note, performance varied more widely across different models than with alternative hyperparameter tuning settings within each model.

Table 7 shows the resulting RF confusion matrices. After class rebalance, the model correctly predicts company success in 82.1%, with a false-positive rate of 10.4% (i.e. companies predicted to succeed, but which fail), and a false-negative rate of 7.5% (companies predicted to fail, but which succeed). In this business context, this would translate into a precision for success predictions of 79.5%, meaning that on average 1 out of 5 companies approached as considered potentially successful in the future would represent failed investments. This compares with 64.7% if no classification model was employed (i.e. baseline model precision for class 1). Although imperfect, a similar approach could be used to target customer acquisition efforts to more suitable companies, potentially leading to important improvements in resource allocation and return-on-investment.

# Solution deployment and limitations

Successful deployment of this solution would require integrating the model into user-friendly applications to be used by company employees without direct input from ML programmers/engineers (by developing bespoke programming code, or using API interfaces to connect to WEKA or other off-the-shelf modelling solutions). This could include the possibility of filtering for companies at specific stages or in specific sectors, with the best possible accuracy presented after retraining of candidate models in each specific data split. Deployment should also allow integration of real or near real-time data feeds given the highly dynamic nature of this sector. Finally, continuous model performance monitoring should be undertaken to account for the impact of changes in the external landscape on likely customer success, namely new regulation, legislation, fundraising environment, or market trends.

Some limitations should be discussed. First, the dataset used only provides a proof-of-concept as it is focused on US companies only, is not recent, and included very limited information on how it was assembled or the expected meaning of some attributes. Deployment of this solution would first require collecting relevant data and developing a model trained on that data, which can be achieved via web-scraping from open online sources (and eventually collection of proprietary data). The data used had high class imbalanced, and skewed towards the class of interest, but this was addressed by applying SMOTE to the underlying data and comparing model performance before and after (leading to accuracy improvements of approximately 1-5%).

Second, model accuracy could have been further improved by more extensive or targeted data collection, and more detailed or exhaustive exploratory data analysis and feature engineering techniques (including formal assessments of collinearity, and further attribute selection refinement). However, this means that the accuracy found here is an underestimation rather than overestimation of the maximum possible model performance. Likewise, other more advanced ML models could also have been employed, such as artificial neural networks, but these were outside the scope of this proof-of-concept.

# Conclusion

This proof-of-concept exercise outlined the steps required for successful development and deployment of an AI-based solution for classification within a business context. The model trained using publicly-available data yielded high accuracy for predicting future start-up success, highlighting the potential added value for company returns if a similar solution were developed using bespoke data collection procedures, further refined with more advanced feature engineer and modelling techniques, and successfully integrated within existing business workflows.

# Tables and figures

## Table 1 – Dataset structure and descriptive summary

| **Attribute name** | **Description** | **Attribute group** | **Retain in model** | **Summary statistics**  **(for variables retained in the model)** | **Transformations applied**  **(for variables retained in the model)** |
| --- | --- | --- | --- | --- | --- |
| unnamed: 0 | Unique identifier | Identifier |  |  |  |
| state\_code | US state code | Location |  | Number of distinct: 35 |  |
| latitude | Latitude | Location |  |  |  |
| longitude | Longitude | Location |  |  |  |
| zip\_code | ZIP code | Location |  |  |  |
| id | Unknown | Identifier |  |  |  |
| city | City | Location |  | Number of distinct: 221 |  |
| unnamed: 6 | City + ZIP code | Location |  |  |  |
| name | Company name | Identifier |  |  |  |
| labels | 0-1 label (based on *status*) | Outcome | Yes | 0: 326 (35.3%)  1: 597 (64.7%) | Numeric to nominal |
| founded\_at | Date company founded | Fundraising/  milestones |  | Minimum: 01/01/1984  Maximum: 16/04/2013 |  |
| closed\_at | Date company closed | Outcome |  | Minimum:  01/01/2001  Maximum:  30/10/2013 |  |
| first\_funding\_at | Date of first funding round | Fundraising/  milestones |  | Minimum:  01/01/2000  Maximum:  20/11/2013 |  |
| last\_funding\_at | Date of last funding round | Fundraising/  milestones |  | Minimum:  01/01/2001  Maximum:  20/11/2013 |  |
| age\_first\_funding\_at | Age of company at first funding round (calculated from founding date) | Fundraising/  milestones | Yes | Minimum: -9.047  Maximum: 21.896  Mean: 2.236  SD: 2.51 | Nominal to numeric |
| age\_last\_funding\_at | Age of company at last funding round (calculated from founding date) | Fundraising/  milestones | Yes | Minimum: -9.047  Maximum: 21.896  Mean: 3.931  SD: 2.968 |  |
| age\_first\_milestone\_year | Age of company at first milestone (calculated from founding date); NB definition of milestones is unknown | Fundraising/  milestones | Yes | Minimum: -14.17  Maximum: 24.685  Mean: 3.055  SD: 2.977 |  |
| age\_last\_milestone\_year | Age of company at last milestone (calculated from founding date); NB definition of milestones is unknown | Fundraising/  milestones | Yes | Minimum: -7.005  Maximum: 24.685  Mean: 4.754  SD: 3.212 |  |
| relationships | Number of relationships established with other businesses/investors | Fundraising/  milestones | Yes | Minimum: 0  Maximum: 63  Mean: 7.71  SD: 7.266 |  |
| funding\_rounds | Number of funding rounds undertaken | Fundraising/  milestones | Yes | Minimum: 1  Maximum: 10  Mean: 2.311  SD: 1.391 | Ordinal to numeric |
| funding\_total\_usd | Total funding secured (in USD) | Fundraising/  milestones | Yes | Minimum: 11,000  Maximum: 5,700,000,000  Mean: 25,419,749.092  SD: 189,634,364.489 | Log transformed |
| milestones | Number of milestones achieved (NB definition of milestones is unknown) | Fundraising/  milestones | Yes | Minimum: 0  Maximum: 8  Mean: 1.842  SD: 1.323 |  |
| state\_code.1 | Repeat from state\_code | Location |  |  |  |
| is\_CA | One-hot encoding for location in California | Location | Yes | 1: 487 (52.8%) | Numeric to nominal |
| is\_NY | One-hot encoding for location in New York | Location | Yes | 1: 106 (11.5%) | Numeric to nominal |
| is\_MA | One-hot encoding for location in Massachusetts | Location | Yes | 1: 83 (9.0%) | Numeric to nominal |
| is\_TX | One-hot encoding for location in Texas | Location | Yes | 1: 42 (4.6%) | Numeric to nominal |
| is\_otherstate | One-hot encoding for location in other states | Location | Yes | 1: 204 (22.1%) | Numeric to nominal |
| category\_code | Sector category | Sector |  |  |  |
| is\_software | One-hot encoding for software sector | Sector | Yes | 1: 153 (16.6%) | Numeric to nominal |
| is\_web | One-hot encoding for web sector | Sector | Yes | 1: 144 (15.6%) | Numeric to nominal |
| is\_mobile | One-hot encoding for mobile sector | Sector | Yes | 1: 79 (8.6%) | Numeric to nominal |
| is\_enterprise | One-hot encoding for enterprise sector | Sector | Yes | 1: 73 (7.9%) | Numeric to nominal |
| is\_advertising | One-hot encoding for advertising sector | Sector | Yes | 1: 62 (6.7%) | Numeric to nominal |
| is\_gamesvideo | One-hot encoding for videogames sector | Sector | Yes | 1: 52 (5.6%) | Numeric to nominal |
| is\_ecommerce | One-hot encoding for e-commerce sector | Sector | Yes | 1: 25 (2.7%) | Numeric to nominal |
| is\_biotech | One-hot encoding for biotech sector | Sector | Yes | 1: 34 (3.7%) | Numeric to nominal |
| is\_consulting | One-hot encoding for consulting sector | Sector | Yes | 1: 3 (0.3%) | Numeric to nominal |
| is\_othercategory | One-hot encoding for other sectors | Sector | Yes | 1: 298 (32.3%) | Numeric to nominal |
| object\_id | Repeat from *id* | Identifier |  |  |  |
| has\_VC | One-hot encoding for VC funding | Fundraising/  milestones | Yes | 1: 301 (32.6%) | Numeric to nominal |
| has\_angel | One-hot encoding for angel funding | Fundraising/  milestones | Yes | 1: 235 (25.5%) | Numeric to nominal |
| has\_roundA | One-hot encoding for round A funding | Fundraising/  milestones | Yes | 1: 469 (50.8%) | Numeric to nominal |
| has\_roundB | One-hot encoding for round B funding | Fundraising/  milestones | Yes | 1: 362 (39.2%) | Numeric to nominal |
| has\_roundC | One-hot encoding for round C funding | Fundraising/  milestones | Yes | 1: 215 (23.3%) | Numeric to nominal |
| has\_roundD | One-hot encoding for round D funding | Fundraising/  milestones | Yes | 1: 92 (10.0%) | Numeric to nominal |
| avg\_participants | Average number of participating investors in each funding round | Fundraising/  milestones | Yes | Min: 1  Max: 16  Mean: 2.839  SD: 1.875 |  |
| is\_top500 | Is it a Top 500 company? | Sector | Yes | 1: 747 (80.95) | Numeric to nominal |
| status | Current status (source for *label* and based on *closed\_at*) | Outcome |  |  |  |

## Table 2 – Decision tree hyperparameter tuning (before SMOTE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hyperparameter** | **Hyperparameter specification** | **Accuracy (%)** | **F-score**  **(weighted average)** | **AUROC** |
| **Default model** | **Default model** | 72.3 | 0.715 | 0.675 |
| **Confidence factor** | **0.10** | 74.5 | 0.734 | 0.711 |
| **0.05\*** | 76.0 | 0.746 | 0.726 |
| **0.01** | 75.8 | **0.750** | 0.725 |
| **Minimum number of objects per leaf** | **5** | 75.0 | 0.740 | 0.727 |
| **10\*** | **76.1** | 0.748 | 0.730 |
| **15** | 74.8 | 0.736 | 0.735 |
| **20** | 75.6 | 0.743 | **0.746** |
| **30** | 75.8 | 0.745 | 0.744 |
| **Reduced error pruning** | **Yes** | 75.0 | 0.739 | 0.712 |

Legend: Tuning performed sequentially using the best specification identified for each hyperparameter. Asterisks identify altered hyperparameters versus the default model, and the specification chosen. Cells highlighted in bold show the maximum across each performance metric. Default model specifications are as follows: confidence factor 0.25, minimum number of objects 2, no reduced-error pruning. AUROC – Area-under-the-Receiver-Operator-Curve; SMOTE - Synthetic Minority Oversampling Technique

## Table 3 – Random forest hyperparameter tuning (before SMOTE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hyperparameter** | **Hyperparameter specification** | **Accuracy (%)** | **F-score**  **(weighted average)** | **AUROC** |
| **Default model** | **Default model** | **79.3** | **0.785** | 0.820 |
| **Break ties randomly** | **Yes** | 78.5 | 0.777 | 0.816 |
| **Maximum tree depth** | **2** | 75.5 | 0.720 | 0.814 |
| **5** | 78.8 | 0.783 | **0.827** |
| **100** | 78.5 | 0.777 | 0.816 |
| **200** | 78.6 | 0.777 | 0.816 |
| **500** | 78.6 | 0.777 | 0.816 |
| **Number of randomly chosen attributes** | **1** | 77.1 | 0.771 | 0.797 |
| **10** | 78.6 | 0.777 | 0.817 |
| **30** | 77.9 | 0.769 | 0.807 |
| **Number of random trees in forest** | **50** | 78.2 | 0.775 | 0.820 |
| **200** | 78.7 | 0.778 | 0.823 |

Legend: No alternative hyperparameter configurations were selected. Cells highlighted in green show the maximum across each performance metric. Default model specifications are as follows: ties not broken randomly, unlimited maximum tree depth, 0 randomly chosen attributes, 100 random trees. AUROC – Area-under-the-Receiver-Operator-Curve; SMOTE - Synthetic Minority Oversampling Technique

## Table 4 – k-nearest neighbours hyperparameter tuning (before SMOTE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hyperparameter** | **Hyperparameter specification** | **Accuracy (%)** | **F-score**  **(weighted average)** | **AUROC** |
| **Default model** | **Default model** | 61.8 | 0.612 | 0.581 |
| **Number of neighbours (k)** | **2** | 56.3 | 0.573 | 0.596 |
| **3** | 62.9 | 0.605 | 0.586 |
| **4** | 59.8 | 0.599 | 0.599 |
| **5** | 66.5 | 0.630 | 0.605 |
| **6** | 65.5 | 0.640 | 0.613 |
| **7** | **67.5** | 0.638 | 0.610 |
| **8** | 66.2 | 0.640 | 0.618 |
| **9** | 67.2 | **0.672** | 0.627 |
| **10\*** | **67.5** | 0.648 | 0.631 |
| **15** | 67.0 | 0.621 | **0.649** |
| **Distance weighting** | **1/distance** | 66.5 | 0.631 | 0.631 |
| **1-distance** | 67.0 | 0.629 | 0.631 |

Legend: Tuning performed sequentially using the best specification identified for each hyperparameter. Asterisks identify altered hyperparameters versus the default model, and the specification chosen. Cells highlighted in bold show the maximum across each performance metric. Default model specifications are as follows: 1 neighbour, no distance weighting. AUROC – Area-under-the-Receiver-Operator-Curve; SMOTE - Synthetic Minority Oversampling Technique

## Table 5 – Support vector machine hyperparameter tuning (before SMOTE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hyperparameter** | **Hyperparameter specification** | **Accuracy (%)** | **F-score**  **(weighted average)** | **AUROC** |
| **Default model** | **Default model** | 74.5 | 0.733 | 0.689 |
| **Build calibration models** | **Yes\*** | 74.6 | 0.738 | 0.785 |
| **Complexity parameter** | **2\*** | **74.9** | **0.741** | 0.786 |
| **3** | 74.2 | 0.735 | **0.788** |
| **5** | 74.1 | 0.735 | 0.788 |
| **Training data filter** | **Standardise** | 74.1 | 0.735 | 0.785 |
| **No normalisation/**  **standardisation** | 74.1 | 0.735 | 0.786 |
| **Kernel used** | **Normalised polykernel** | 73.2 | 0.718 | 0.777 |
| **Puk** | 66.7 | 0.666 | 0.683 |
| **RBF** | 70.7 | 0.680 | 0.757 |

Legend: Tuning performed sequentially using the best specification identified for each hyperparameter. Asterisks identify altered hyperparameters versus the default model, and the specification chosen. Cells highlighted in green show the maximum across each performance metric. Default model specifications are as follows: no calibration models built, complexity parameter 1.0, training data normalised, polykernel. AUROC – Area-under-the-Receiver-Operator-Curve; SMOTE - Synthetic Minority Oversampling Technique

## Table 6 – Comparative summary of performance metrics across candidate algorithms

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **SMOTE** | **Accuracy (%)** | **Precision** | | **Recall** | | **F1-score** | | **AUROC** |
| Target class (1) | Weighted average | Target class (1) | Weighted average | Target class (1) | Weighted average |
| **Baseline accuracy** *(rules.ZeroR)* | **No** | 64.7 | 0.647 | N/A | 1.00 | 0.647 | N/A | 0.786 | 0.495 |
| **Yes** | 52.2 | N/A | N/A | 0 | 0.522 | N/A | N/A | 0.497 |
| **Logistic regression**  *(functions.logistic)* | **No** | 74.0 | 0.773 | 0.773 | 0.846 | 0.740 | 0.808 | 0.734 | 0.786 |
| **Yes** | 79.6 | 0.761 | 0.799 | 0.836 | 0.796 | 0.796 | 0.796 | 0.889 |
| **Naïve Bayes**  *(bayes.NaiveBayes)* | **No** | 72.7 | 0.795 | 0.729 | 0.779 | 0.727 | 0.787 | 0.728 | 0.790 |
| **Yes** | 74.5 | 0.732 | 0.745 | 0.745 | 0.745 | 0.733 | 0.745 | 0.824 |
| **Decision tree**  *(trees.J48)* | **No** | 76.1 | 0.770 | 0.756 | 0.898 | 0.761 | 0.829 | 0.748 | 0.730 |
| **Yes** | 77.1 | 0.754 | 0.771 | 0.774 | 0.771 | 0.764 | 0.771 | 0.789 |
| **Random Forest**  *(trees.RandomForest)* | **No** | 79.3 | **0.801** | 0.791 | **0.905** | 0.793 | **0.850** | 0.785 | 0.820 |
| **Yes** | **82.1** | 0.795 | **0.822** | 0.843 | **0.821** | 0.818 | **0.821** | **0.899** |
| **K-Nearest Neighbours**  *(lazy.IBk)* | **No** | 67.5 | 0.701 | 0.655 | 0.866 | 0.675 | 0.775 | 0.648 | 0.631 |
| **Yes** | 71.2 | 0.736 | 0.715 | 0.620 | 0.712 | 0.673 | 0.709 | 0.804 |
| **Support Vector Machine**  *(functions.SMO)* | **No** | 74.9 | 0.774 | 0.742 | 0.863 | 0.749 | 0.816 | 0.741 | 0.786 |
| **Yes** | 79.7 | 0.771 | 0.798 | 0.817 | 0.797 | 0.793 | 0.797 | 0.890 |

Legend: The WEKA model implementation used to run each algorithm is outlined in italic. The target class (label 1) represents successful acquisition. Cells highlighted in green show the maximum for each performance metric. AUROC – Area-under-the-Receiver-Operator-Curve; SMOTE - Synthetic Minority Oversampling Technique

## Table 7 – Confusion matrix for the optimal model (random forest)

A - Before SMOTE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Observed** | | |
| **Success** | **Fail** | **Total** |
| **Predicted** | **Success** | **540 (58.5%)** | **134 (14.5%)** | 674 (73.0%) |
| **Fail** | **57 (6.2%)** | **192 (20.8%)** | 249 (27.0%) |
| **Total** | 597 (64.7%) | 326 (35.3%) | 923 (100%) |

Total accuracy: 79.3%

B - After SMOTE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Observed** | | |
| **Success** | **Fail** | **Total** |
| **Predicted** | **Success** | **503 (40.3%)** | **130 (10.4%)** | 633 (50.7%) |
| **Fail** | **94 (7.5%)** | **522 (41.8%)** | 616 (49.3%) |
| **Total** | 597 (47.8%) | 652 (52.2%) | 1249 (100%) |

Total accuracy: 82.1%

Legend: Cells highlighted in green show correct predictions, those highlighted in orange show false-positives (i.e. model predicts success but company failed, or “failed hunches”), and those in yellow show false-negatives (i.e. model predicts failure but company succeeded, or “missed opportunities”). SMOTE - Synthetic Minority Oversampling Technique

# References

Bell, J. (2020) *Machine Learning: Hands-On for Developers and Technical Professionals*. 2nd edn. Chichester: Wiley.

Bessen, J., Impink, S.M., Reichensperger, L. & Seamans, R. (2022) ‘The role of data for AI startup growth’, *Research Policy*, 51(5), p. 104513. Available from: https://doi.org/10.1016/j.respol.2022.104513.

Buchanan, B.G. (2019) *Artificial intelligence in finance*. The Alan Turing Institute. Available from: https://www.turing.ac.uk/sites/default/files/2019-04/artificial\_intelligence\_in\_finance\_-\_turing\_report\_1.pdf (Accessed: 27 June 2024).

Dellermann, D., Lipusch, N., Ebel, P., Popp, K.M. & Leimeister, J.M. (2021) ‘Finding the unicorn: Predicting early stage startup success through a hybrid intelligence method’. arXiv. Available from: https://doi.org/10.48550/ARXIV.2105.03360.

Duboue, P. (2020) ‘The Art of Feature Engineering’, in. Cambridge University Press.

Foody, G.M. (2023) ‘Challenges in the real world use of classification accuracy metrics: From recall and precision to the Matthews correlation coefficient’, *PLOS ONE*. Edited by S. Huang, 18(10), p. e0291908. Available from: https://doi.org/10.1371/journal.pone.0291908.

Frank, E., Hall, M.A. & Witten, I.H. (2016) *The WEKA Workbench*. 4th edn. Morgan Kaufmann.

Potanin, M., Chertok, A., Zorin, K. & Shtabtsovsky, C. (2023) ‘Startup success prediction and VC portfolio simulation using CrunchBase data’. arXiv. Available from: https://doi.org/10.48550/ARXIV.2309.15552.

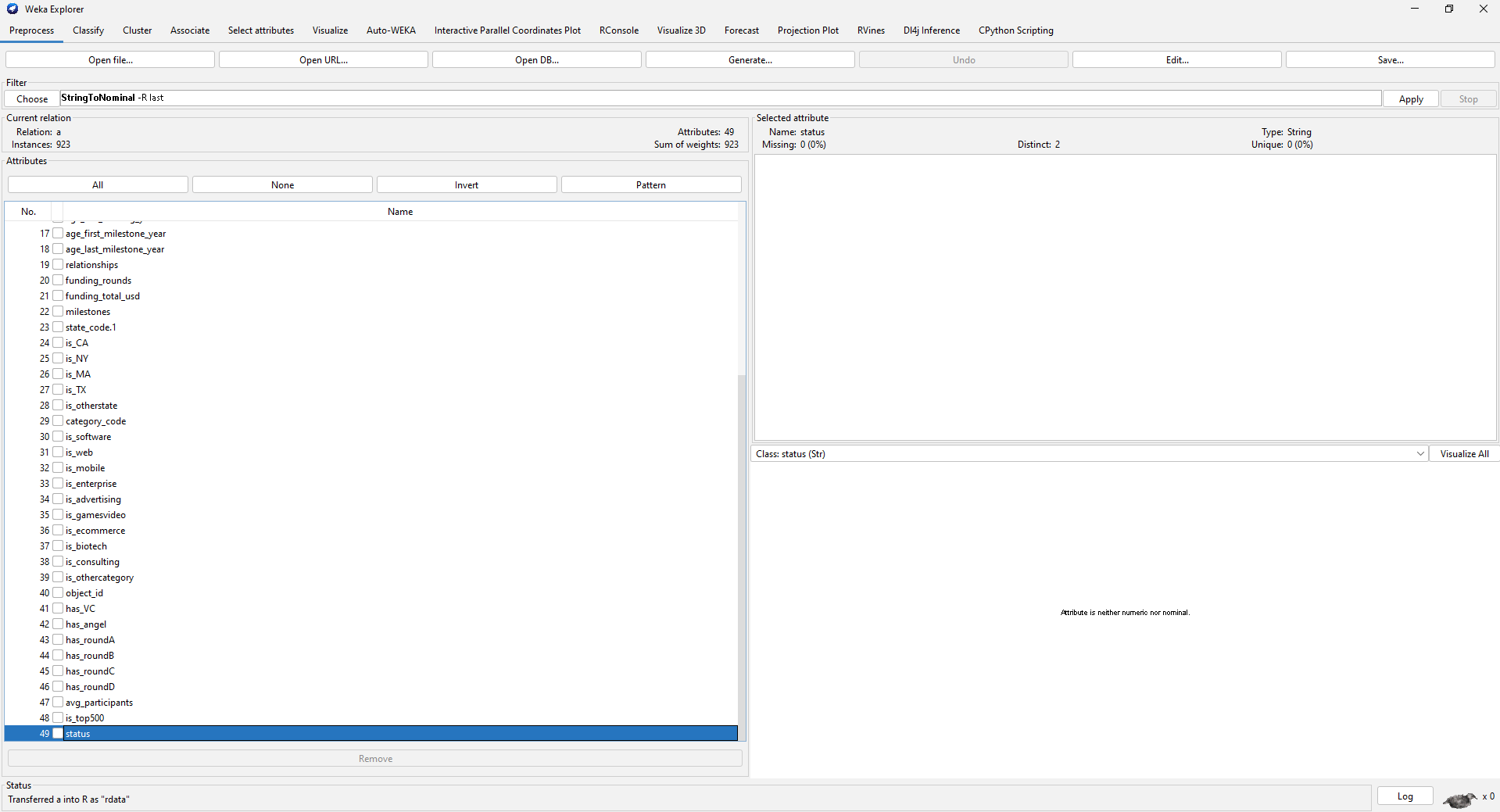
Raschka, S. (2018) ‘Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning’. arXiv. Available from: https://doi.org/10.48550/ARXIV.1811.12808.

Russel, S. & Norvig, P. (2021) *Artificial intelligence: A modern approach*. Global Edition. Pearson Higher Education.

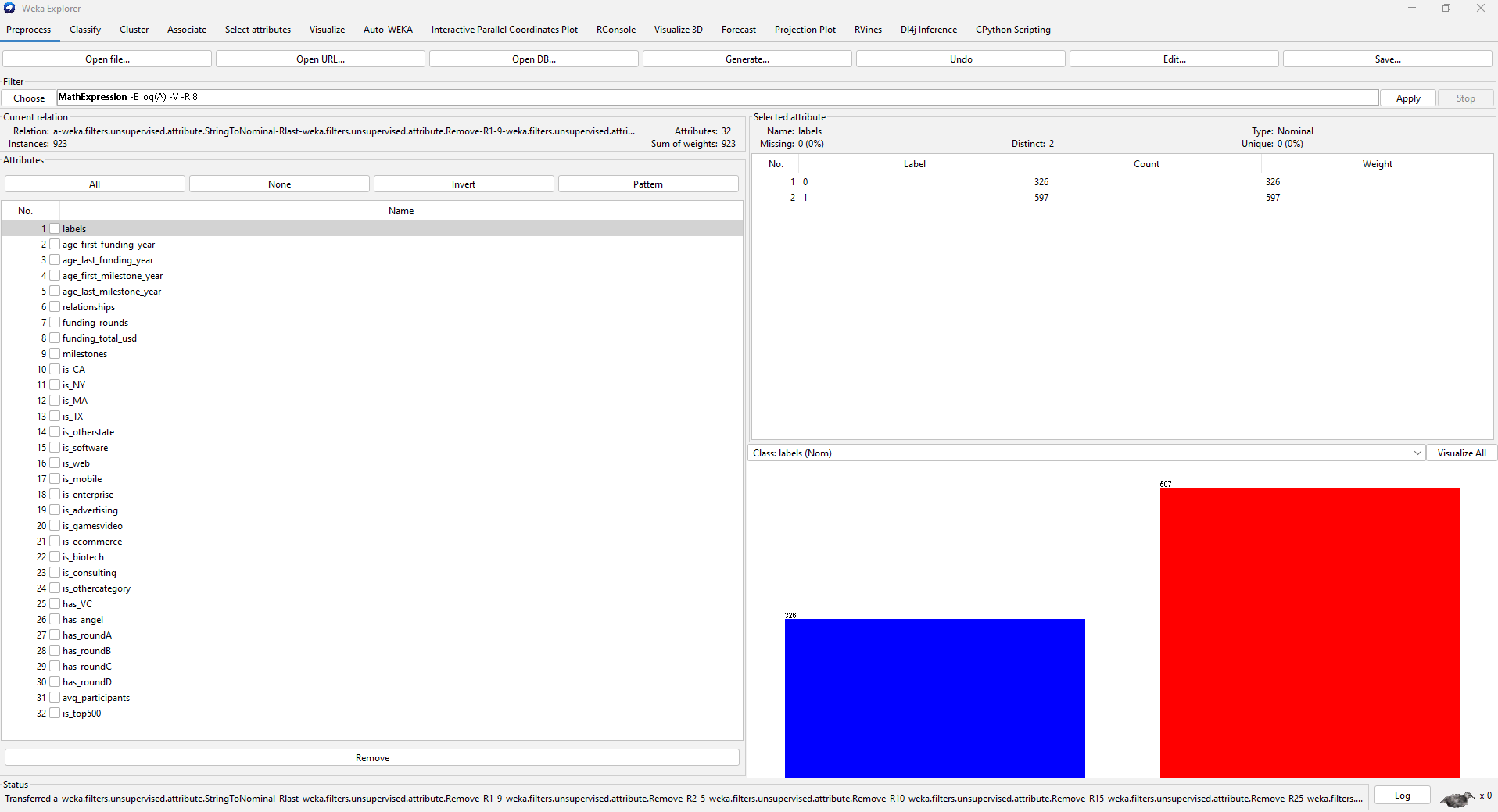
Schröer, C., Kruse, F. & Gómez, J.M. (2021) ‘A Systematic Literature Review on Applying CRISP-DM Process Model’, *Procedia Computer Science*, 181, pp. 526–534. Available from: https://doi.org/10.1016/j.procs.2021.01.199.

# Supplement: WEKA screenshots

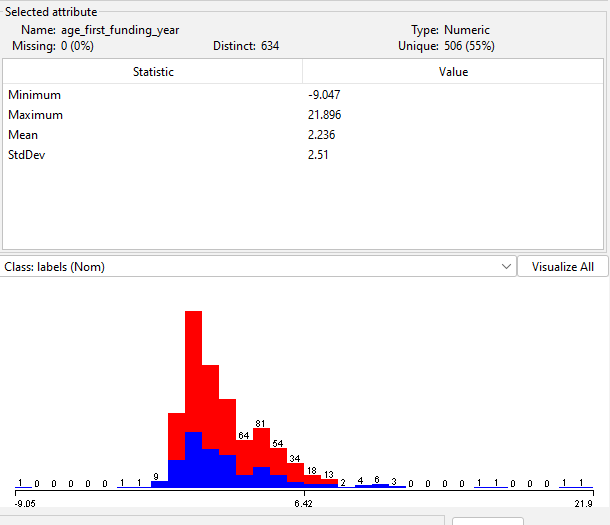
## Figure S1 – Initial data load



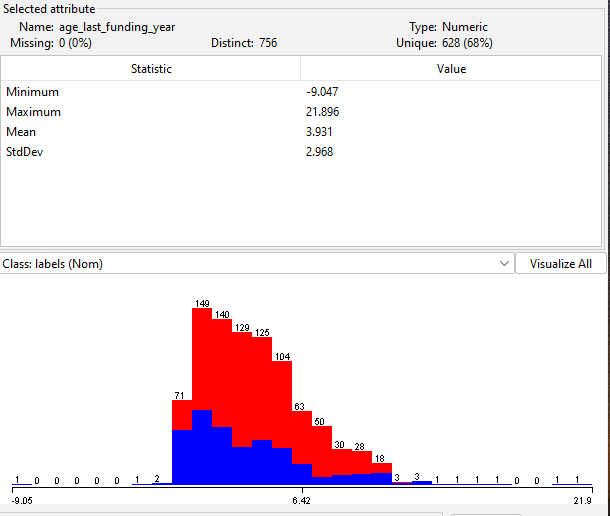
## Figure S2 – Final set of selected attributes and target class overview



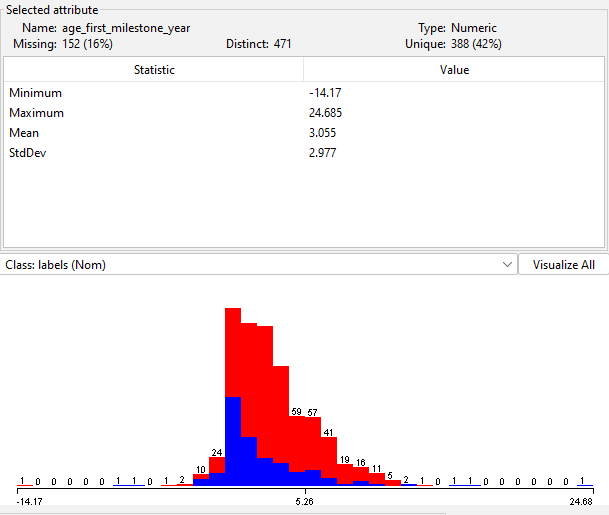
## Figure S3 – Exploratory data analysis: age at first funding round



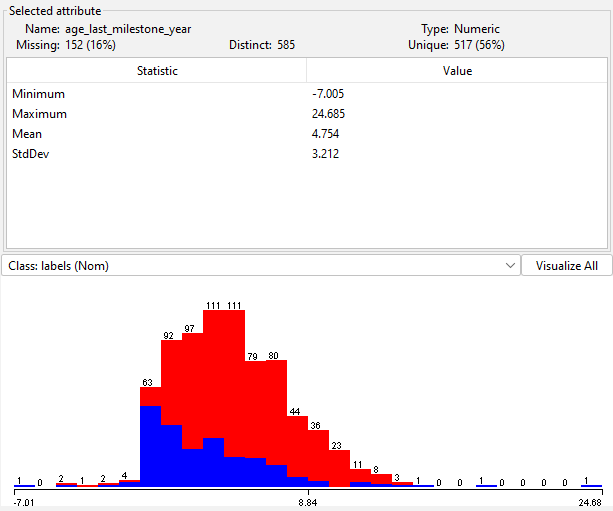
## Figure S4 - Exploratory data analysis: age at last funding round



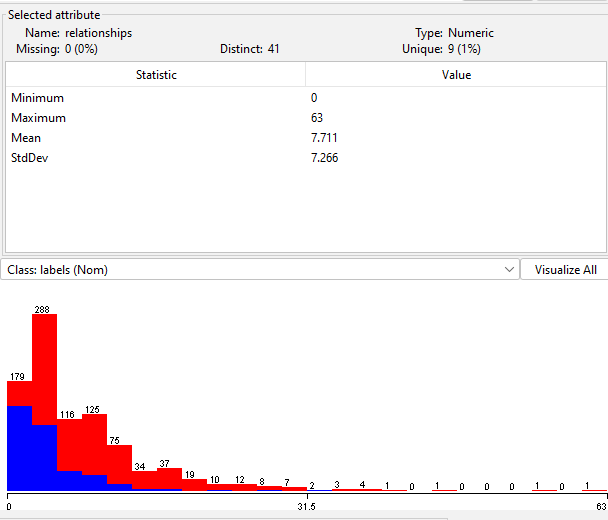
## Figure S5 - Exploratory data analysis: age at first milestone



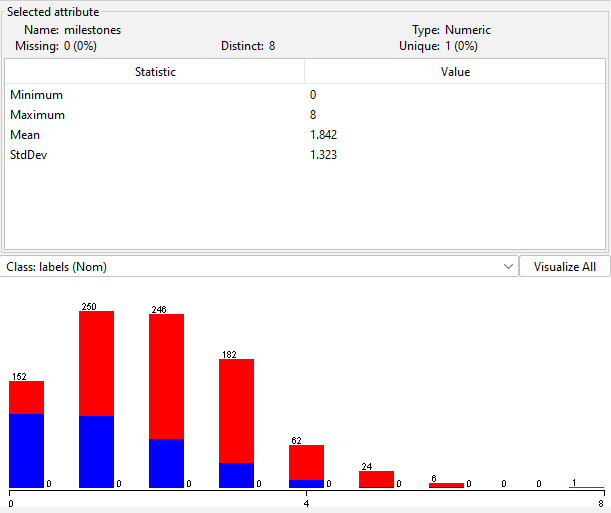
## Figure S6 - Exploratory data analysis: age at last milestone



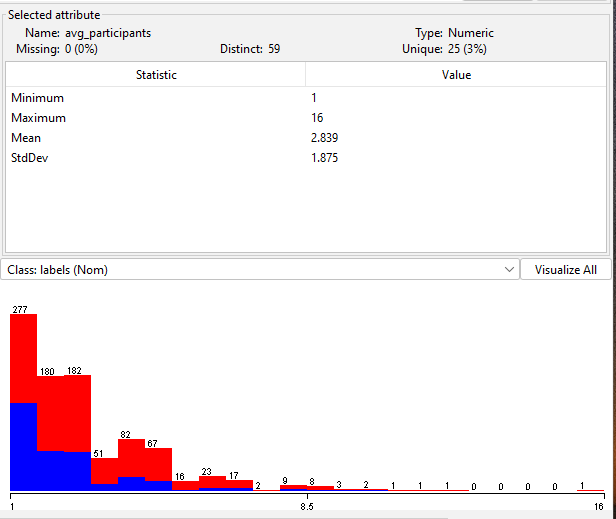
## Figure S7 - Exploratory data analysis: number of relationships



## Figure S8 - Exploratory data analysis: number of milestones achieved

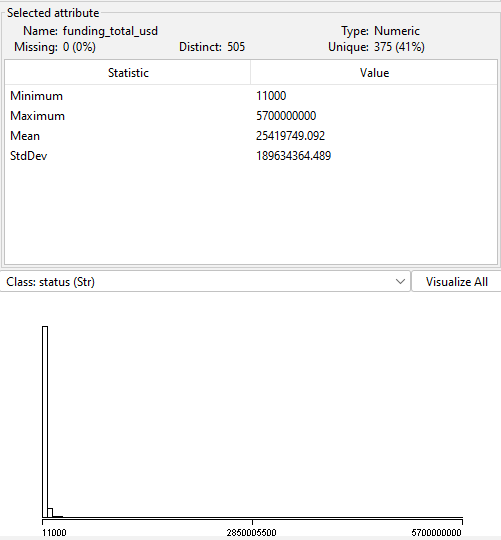


## Figure S9 - Exploratory data analysis: average number of participants in funding rounds

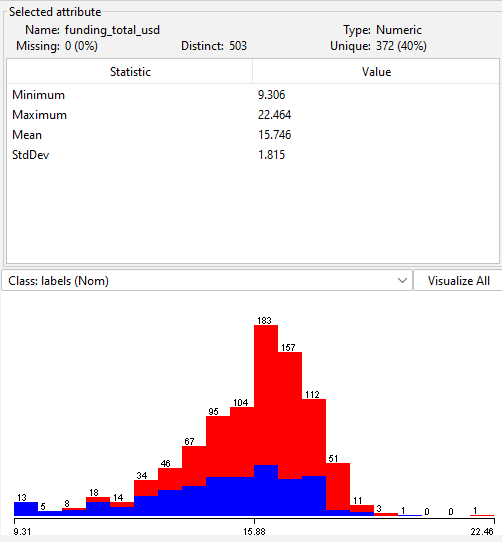


## Figure S10 - Exploratory data analysis: total funding obtained

A – raw data

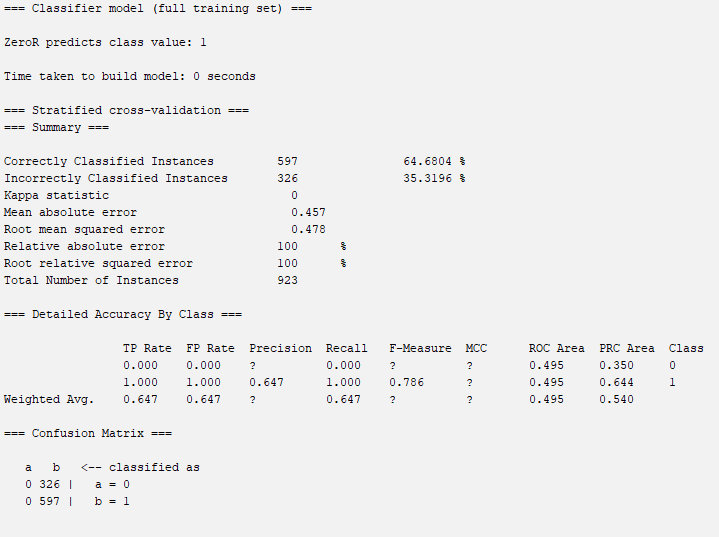


B – after log10 transformation

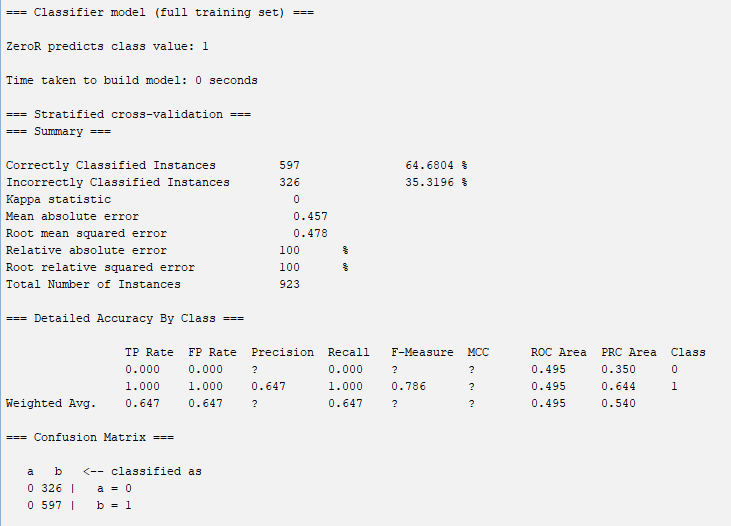


## Figure S11 – Baseline accuracy performance

A – before SMOTE

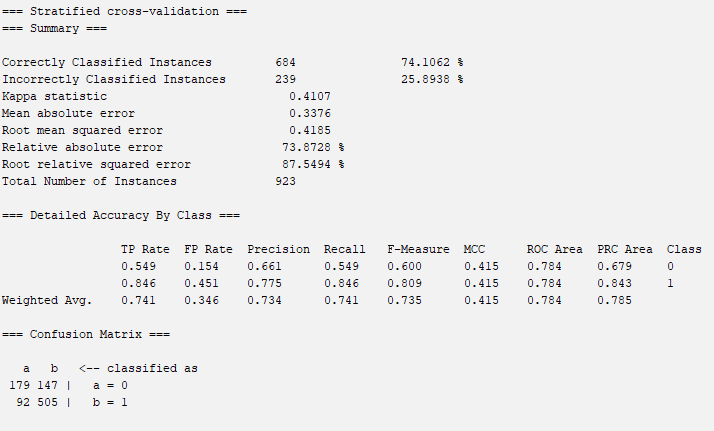


B- after SMOTE

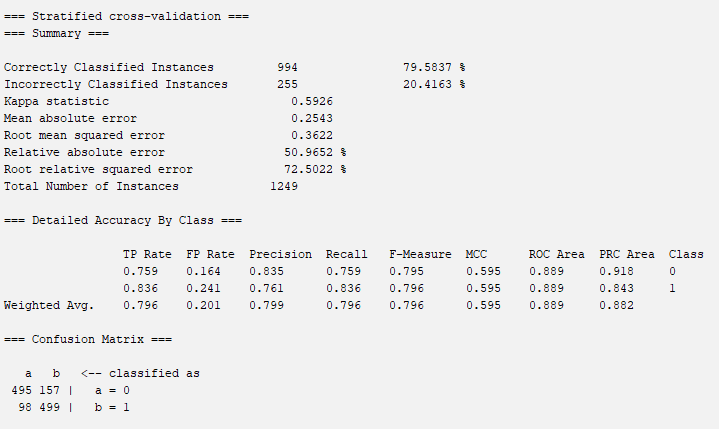


## Figure S12 – Logistic regression performance

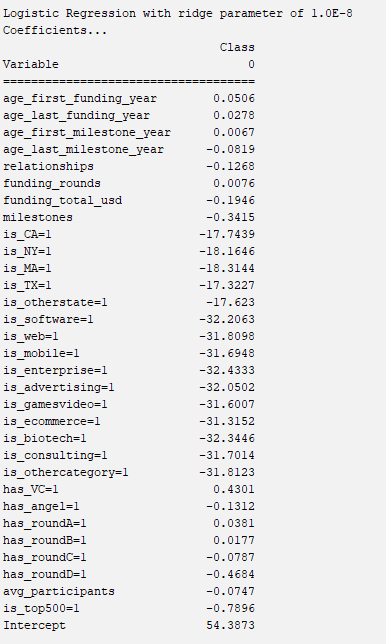
A – before SMOTE



B – after SMOTE

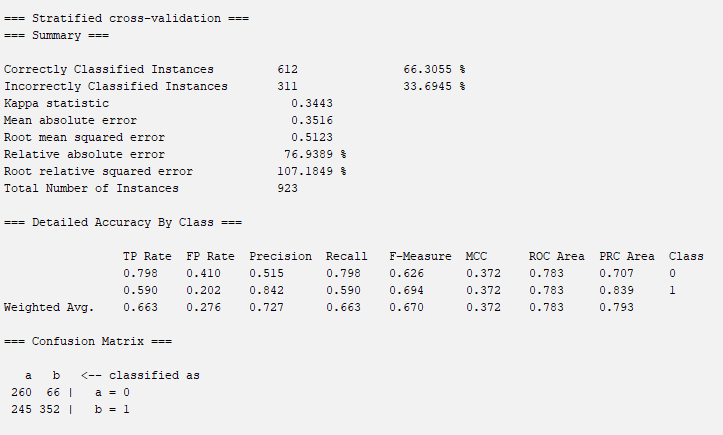


## Figure S13 – Logistic regression parameters (after SMOTE)

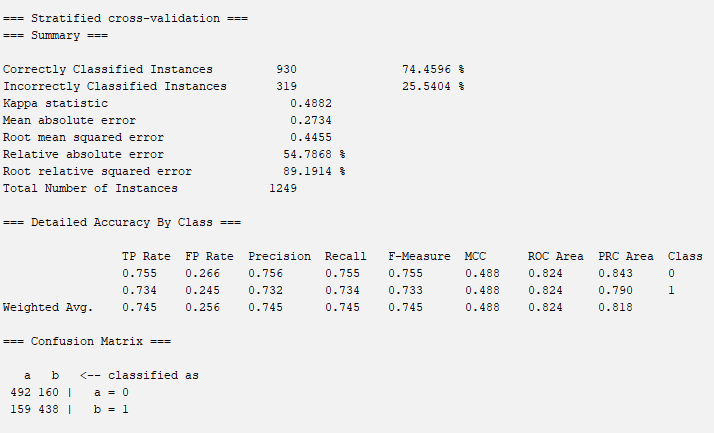


## Figure S14 – Naïve Bayes performance

A – before SMOTE

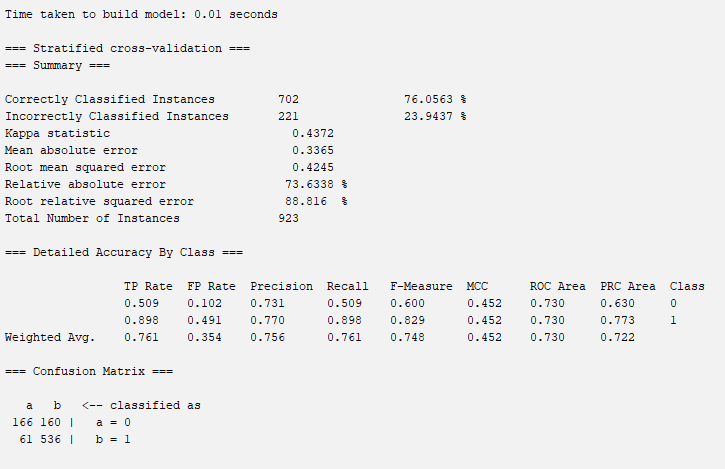


B – after SMOTE

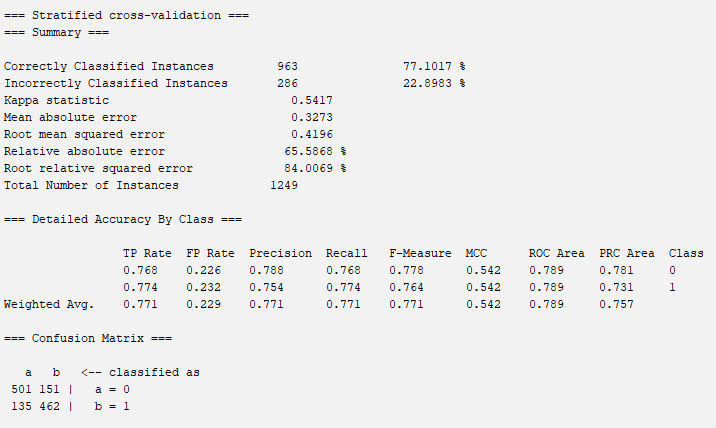


## Figure S15 – Decision tree performance

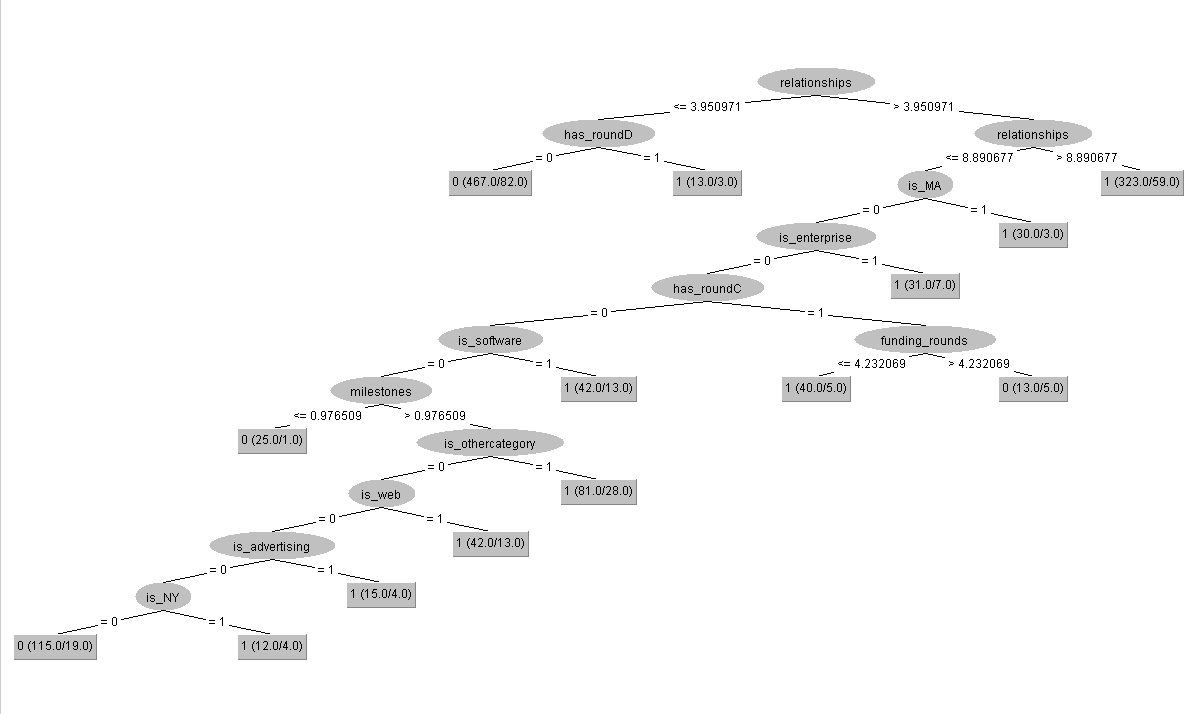
A – before SMOTE



B – after SMOTE

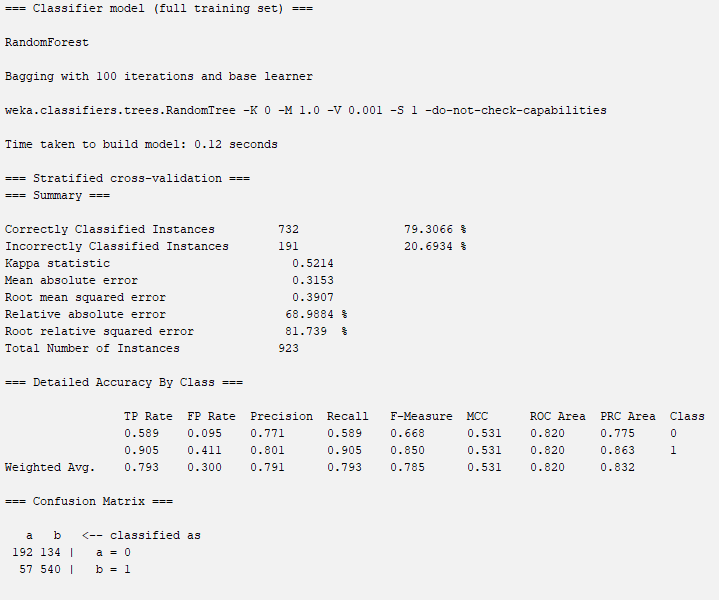


## Figure S16 – Decision tree architecture (after SMOTE)

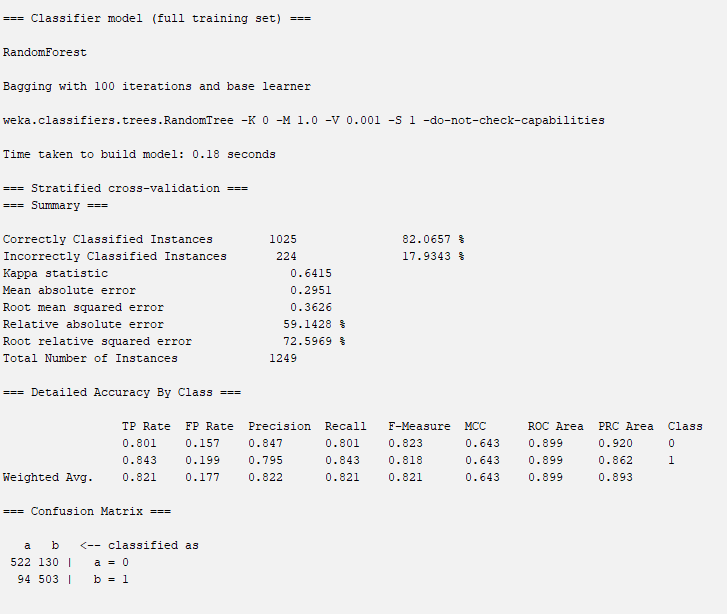


## Figure S17 – Random forest performance

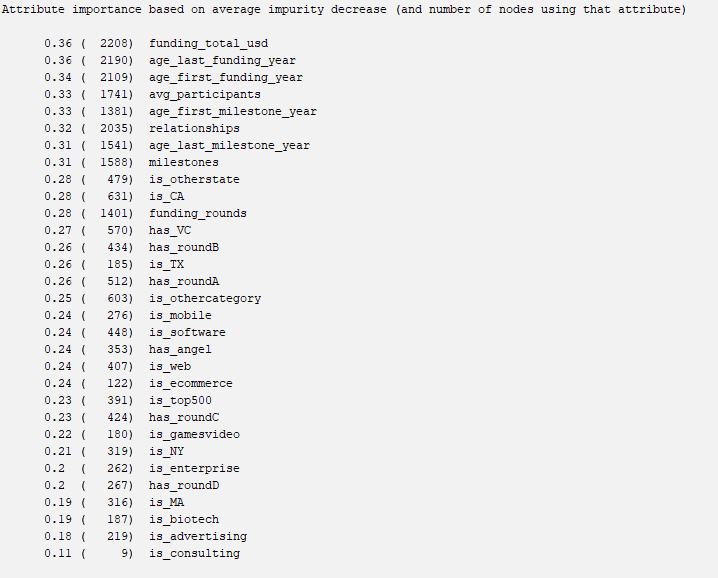
A – before SMOTE



B – after SMOTE

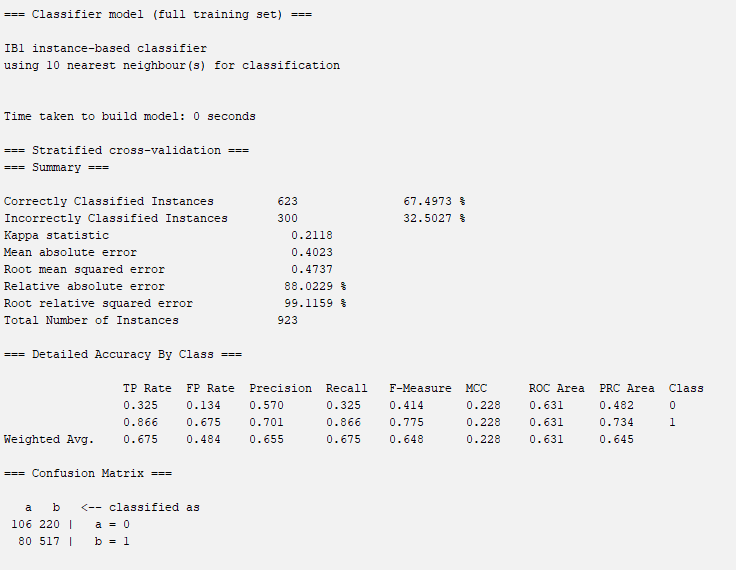


## Figure S18 - Random forest attribute importance (after SMOTE)

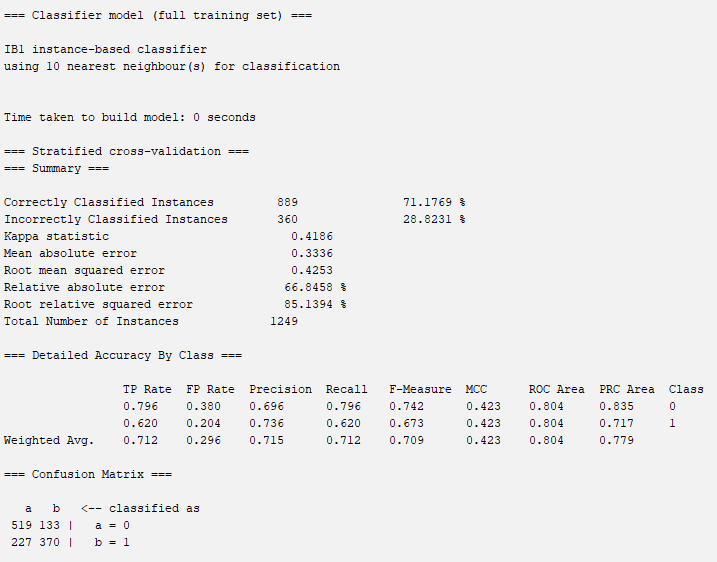


## Figure S19 – K-Nearest Neighbours performance

A – before SMOTE

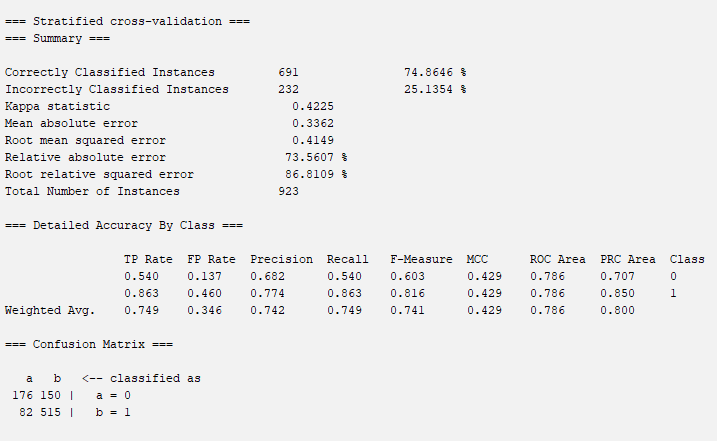


B – after SMOTE

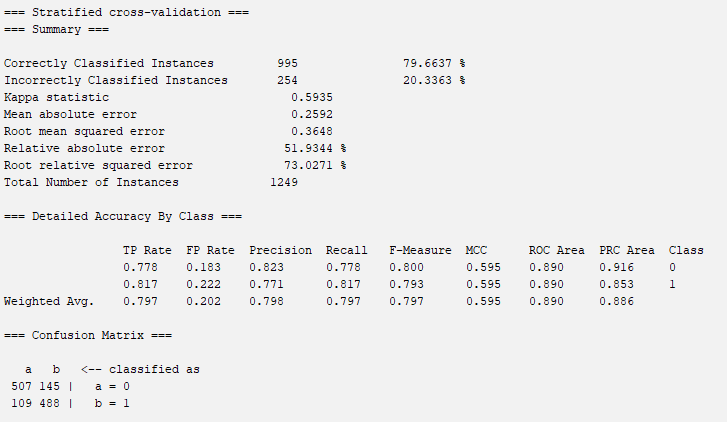


## Figure S20 – Support Vector Machine performance

A – before SMOTE



B – before SMOTE



## Figure S21 – Support Vector Machine weights (after SMOTE)

