Yesika Contreras Duarte

Data Analysis Report

1. Introduction and background

Classification is generally thought of as the problem of assigning a new observation into a predefined group or class with associated features, which is done by utilizing a former dataset (training data) related to that observation [1]. Unbalanced data in classification refers to the unequal distribution of observations for different classes in a feature, most classification algorithms are sensitive to unbalanced classes of the target feature, were the prediction model can be biased toward the majority class. Often the data needs to be balanced for modeling purposes. This is generally accomplished by using under- or over-sampling techniques on the training data. Under-sampling (also known to as down-sampling) refers to a process of randomly selecting a subset of the majority class to match the number of samples per class, however information may be lost from the samples that were not selected. On the other hand, over-sampling (up-sampling) refers to a process of randomly duplicating samples of the minority class to match the number of samples per class. This process carries the risk of overfitting the model [2].

Classification techniques were applied to the dataset provided to build a model that predicts 'var0' using the remaining features as predictors. Additionally, down-sampling and up-sampling approaches were applied to counter unbalanced data.

2. Summary

To build the model that predicts or classifies new observations to the classes of the feature var0, classes '0' or '1', the data set was divided into training data with 70% of the observations, and test data with 30% of the observations. Cross-validation was executed to improve the performance of the models, the technique used was k-folds cross-validation with 10 folds.

The models were built using the Logistic Regression(LR) and Random Forest (RF) techniques under three approaches. First, all the predictor features were transformed intro numeric type and passed into the models with validation. Second, after descriptive analysis of the features, some features were transformed intro categorical type, then this dataset of numeric and categorical features was passed into the models with validation. Third, due that the unbalanced nature of the class distribution in the dataset, i.e. most of the data for the target feature belong to one class, under-sampling and over-sampling were applied to the model after the best model was chosen based on performance in the unbalanced dataset. Metrics were compared to analyze if these sampling approaches improved general performance.

As a result, a Random Forest model with cross-validation and over-sampling in the training set was selected as the model that best fit the target feature var0.

Models were built using R Studio.

3. Exploratory Data Analysis

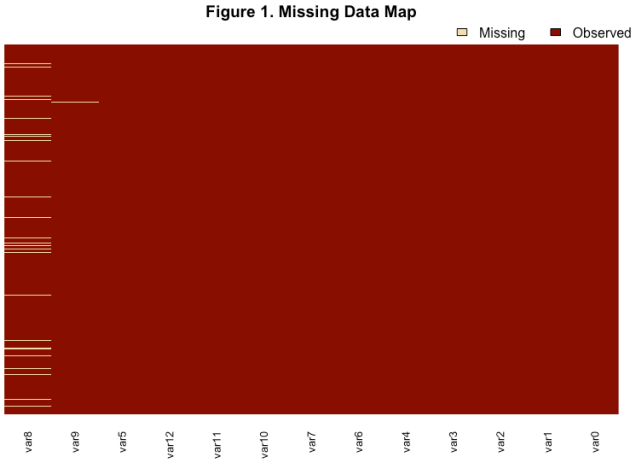
This data set includes 13 features for 100,000 observations. The response features is 'var0', which indicates whether or not the observation presents this feature, having labels '0' or '1'. In this data set, only 2.5% of observations presents this feature (label is 1). Therefore, the target feature represents an unbalance class. Additionally, the data set for this challenge is simulated, and a data dictionary was not provided which can increase the difficulty of feature selection.

- **Cleaning Data: NA values**

Initially no missing values were identified, but after analyzing the content of the features, some observations such as outliers and blanks were replaced to missing values (NA), as showed in the table and represented in figure 1. Since the new NA values represented the 5% of the data, these observations were dropped from the dataset.

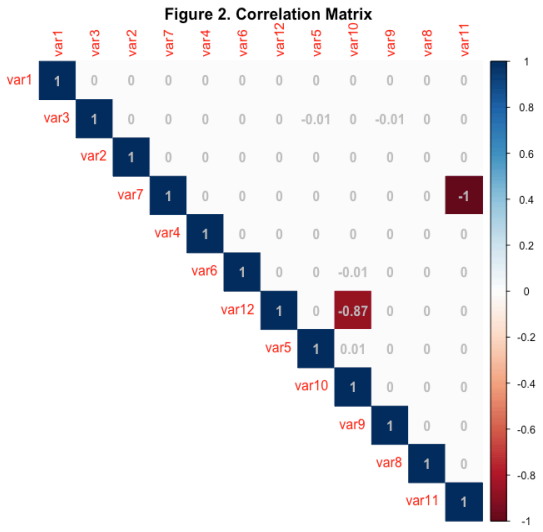
Table 1. Cleaning data

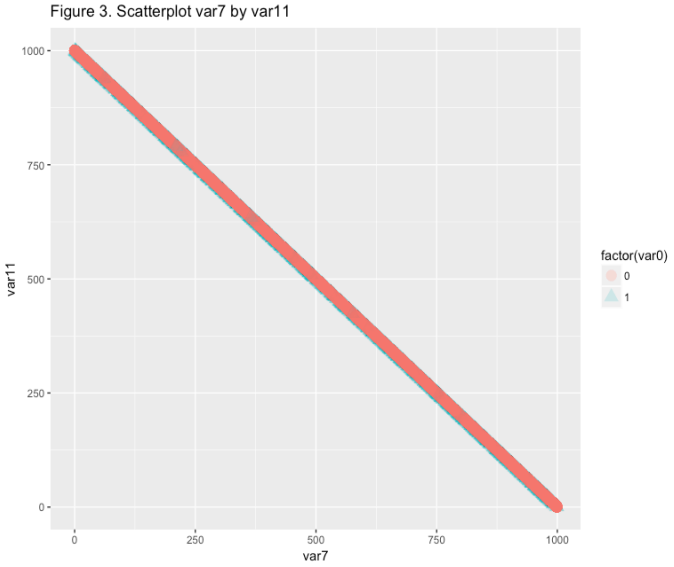
Feature	Value replaced by NA	Observations replaced
Var5	9999	11
Var8	(blank)	4934
Var9	-999	100
Total		5045 = 5% of data



- **Features transformation**

10 features were identified in the initial dataset as numeric and two others as factor. Without having further information of the features it was preferable to build the models using the numeric datatype. To this end, var8 and var10 were transformed into the numeric data type to build the initial models. Figure 2 shows the correlation after transformation, there is no correlation among most of the features. Interestingly, the var7 and var11 (figure 3) are perfectly negatively correlated, and var12 & var10 exhibit a high negative correlation. Multicollinearity, high correlation between predictors, could affects performance in the model or unstable solutions [3]. To minimize the efect of multicollinearity, var7 which was the feature that contribute the most to the correlation was removed.





Observing the data, most of the data falls in class 0 for var0 (figure 4), equivalent to the 97.5% of the observations, which confirms that the dataset is unbalanced. Although the data is unbalanced , some features have the same distribution of data among the class 0 and class 1, as presented in var4 (figure 5).

Figure 4. Barplot Target Feature (var0)

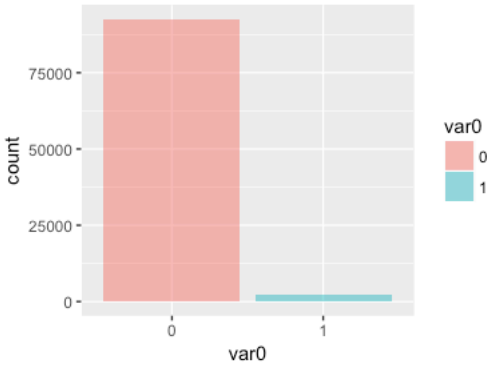
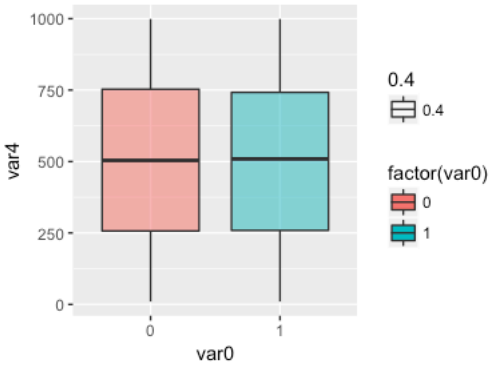


Figure 5. Boxplot var4 by Target Feature



Additionally, to compare model performance, a dataset with numeric and categorical features was generated. Features that could be considered as factors were analyzed. These features were simulated as factors and analyzed for their relationship with the target variable var0. For instance, var2 was transformed into categorical feature becaused it presented variation in the distribution of classes (levels) for the target feature as opposing to var1 that did not present significant changes.

Figure 6. Barplot var2 by factors

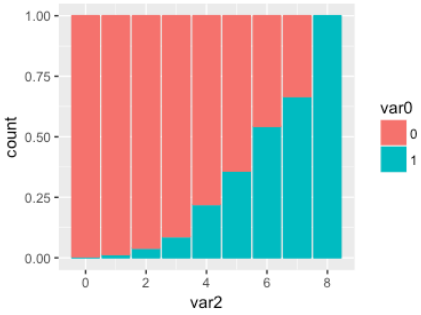
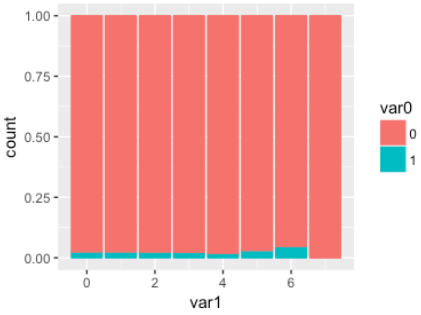


Figure 7. Barplot var1 by factors



4. Models

- Training and test sets

The data set was divided into training data with 70% of the observations, and test data with 30% of the observations. Cross-validation was executed to improve the performance of the models, the technique used was k-folds cross-validation with 10 folds.

- Model(s)

Step 1:

Models were built using the Logistic Regression(LR) and Random Forest (RF) techniques. LR models utilized stepAIC for feature selection. RF models were generated by using 200 trees by split. Models were built with validation. For all the models, the Positive Class is 0, being the reference for the metrics. Below is presented the model label and its description.

Learning: Cross-validation in train set
Lr1: Logistic Regression model with Numeric features
Lr2: Logistic Regression model with Numeric & Categorical features
rf1: Random Forest with model Numeric features
rf2: Random Forest with model Numeric & Categorical features

Step 2:
After running the models in step one, the best model was selected based on the AUC metric. After choosing this model, under-sampling and over-sampling techniques were used to balance the data and improve the overall performance of the model.

Learning: Cross-validation, Down-sampling
rfDown: Random Forest model with Numeric features and down-sampling in train set. Random forest models have the ability to use down-sampling without data loss.

Learning: Cross-validation, Up-sampling
rfUp: Random Forest model with Numeric features and up-sampling in train set.

- Metric(s)

The performance of the models was evaluated under the metrics of accuracy, misclassification rate, sensitivity, specificity, and AUC (area under the curve). Since the target presents unbalance classes, distribution, accuracy, misclassification rate, or sensitivity are not good indicators of performance especially for step 1. On the other hand, AUC and specificity, may provide a more accurate performance indicator. AUC is the probability of correctly classifying two randomly selected users one from each class. Specificity measure the proportion of observations being '1' that were predicted correctly.

5. Findings

After building the initial models (lr1, lr2, rf1, rf2), the model rf1, Random Forest with Numeric features, was selected as the best model. However, to balance the data and a generate a model with better performance classifying both classes of the target feature var0, up- and down-sampling was used in the training data. In addition, cross-validation was used to the newly generated data sets from the training data to provide a honest estimate of the model performance. Surprisingly, the model generated with over-sampling (rfUp) had a perfect performance in all the metrics and it was selected as the best model that fit the var0.

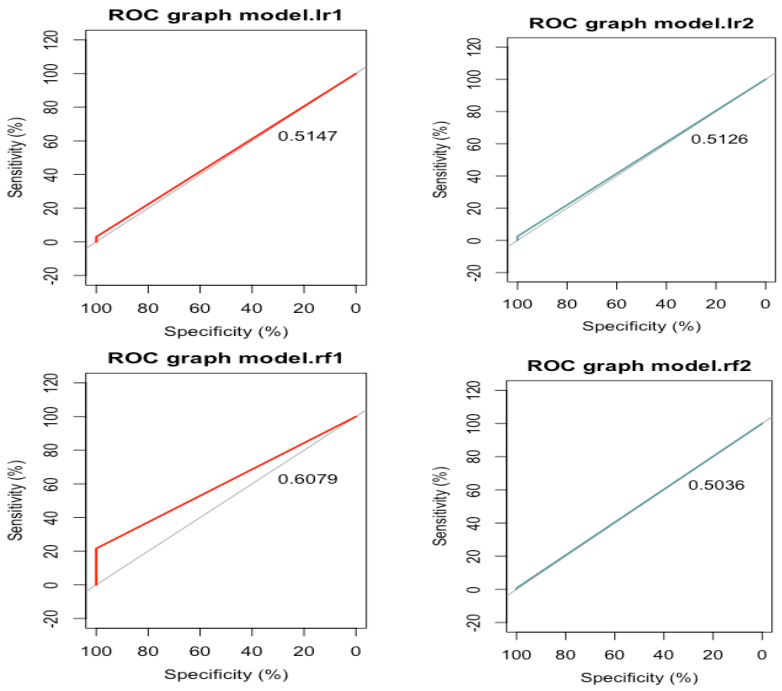
- Comparison of learning systems

The following table shows the result of the metrics by model and the figure depicts different ROC graphs per model.

Table 2. Comparison of learning systems

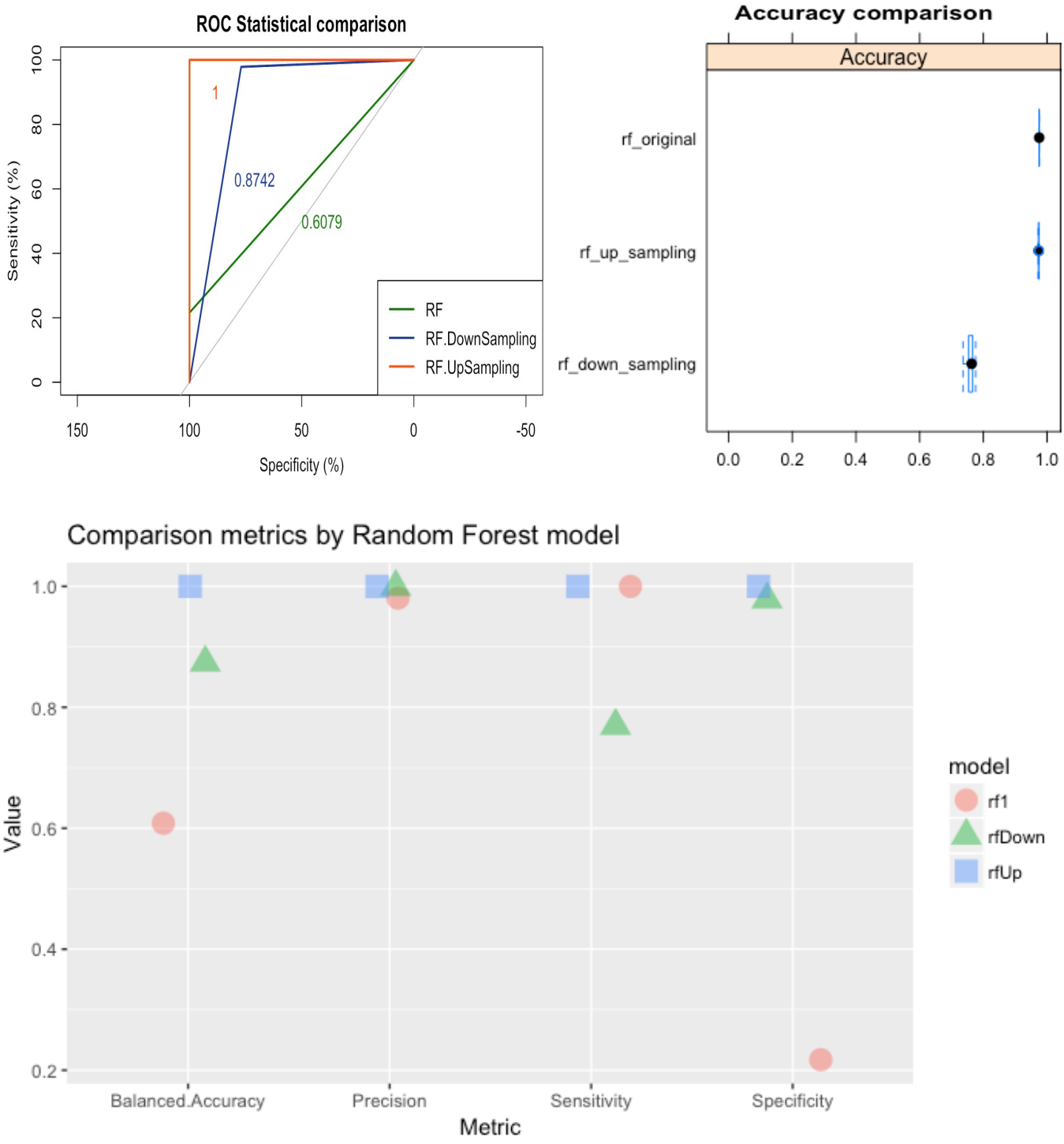
Learning: Cross-validation	Metric				
Model	Accuracy	Misclassification Rate	Sensitivity	Specificity	AUC
lr1	0.9756	0.0244	0.9994	0.0300	0.5197
Lr2	0.9755	0.0245	0.9994	0.0257	0.5126
rf1	0.9807	0.0192	1.0000	0.2171	0.6064
Rf2	0.9756	0.0244	1.0000	0.0071	0.5036
Learning: Cross-validation, Down-sampling	Metric				
Model	Accuracy	Misclassification Rate	Sensitivity	Specificity	AUC
rfDown	0.7744	0.2256	0.76929	0.97857	0.8741
Learning: Cross-validation, Up-sampling	Metric				
Model	Accuracy	Misclassification Rate	Sensitivity	Specificity	AUC
rfUp	1.0000	0.0000	1.0000	1.0000	1.0000

Figure 8. ROC graph by model – unbalance data models



For step 1, the models present better performance when all the features were considered as numeric using cross-validation technique. Comparing different metrics, the specificity presents low values for all the models due to the unbalance class distribution. AUC metric presented better results for Random Forest model rf1, also this model presented better performance in all the metrics. Therefore, model rf1 was selected as the model with best fit in step 1.

Figure 9. Graphs by model –
unbalance data model selected vs. balanced data models



After select rf1 as the model with the best performance, a new Random Forest model was built using down-sample and up-sampling to counter the unbalance class distribution, showing significant improvements in the AUC and Specificity metrics. Although the rfDown model improved in AUC and specificity, the other metrics measured lower values than then other models. The model rfUp presented a prefect performance for all the metrics.

- Model selected

The model that best fit the var0 is rfUp, Random Forest model with Numeric features and up-sampling in train set. Although given the information provided there is no a real way to obtain a correct evaluation of performance, however, AUC was determined to be the best metric to evaluate the performance of the model. This was due to the ability of AUC to show the performance of a model for both classes of var0, which represents a more truthful performance than accuracy for an unbalanced dataset.

This model used resampling: Cross-Validated (10 fold) for a total of 94963 samples, were every sample had accuracy around 0.9737. The final value used for the model was mtry of 6, which is the number of variables randomly sampled as candidates at each split (bagging). The importance of the features in the model is showed in the followed table, were the importance is given in a range from 100 to 0 and calculated based on the mean Gini gain.

Table 3. Importance of the features

Feature	Importance
var2	100
var3	33.003
var4	17.363
var11	16.479
var9	13.527
var1	8.157
var6	6.979
var5	5.942
var10	5.777
var8	3.228
var12	0

6. Code

The R code is provided with the step by step procedure to generate the proposed models, distributed in the following files:

1_cleaning.R: Cleaning and Exploratory Data Analysis, the output of this file is the dataframes.RData file.
2_modelLR.R: A Logistic Regression Model is generated here
3_modelRF.R: A Random Forest Model is generated here
4_modelRFDown.R: A Random Forest Model with down-sampling is generated here
5_modelRFUp.R: A Random Forest Model with up-sampling is generated here
6_modelComparison.R: Comparison among Random Forest Model and Random Forest Model with down-smpling and up-sampling

Input file:
Dataset.csv
Corresponding outputs:
dataframes.RData: cleaned dataframes
model_lr: Logistic Regression model output
model_rf: Random Forest model output
model_rfDown: Random Forest Model with down-sampling model output
model_rfUp: Random Forest Model with up-sampling model output

References

[1] Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. Data Classification: Algorithms and Applications, 37. https://pdfs.semanticscholar.org/310e/a531640728702fce6c743c1dd680a23d2ef4.pdf\

[2] Kaur, P., & Gosain, A. (2018). Comparing the Behavior of Oversampling and Undersampling Approach of Class Imbalance Learning by Combining Class Imbalance Problem with Noise. In ICT Based Innovations (pp. 23-30). Springer, Singapore. https://www.researchgate.net/publication/320160451_Comparing_the_Behavior_of_Oversampling_and_Undersampling_Approach_of_Class_Imbalance_Learning_by_Combining_Class_Imbalance_Problem_with_Noise

[3] Building Predictive Models in R Using the caret Package. http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/BuildingPredictiveModelsR_caret.pdf

Author

Yesika Contreras Duarte
MS Business Analytics UIC