# Credit\_Risk\_Analysis

## Overview

The purpose of the project is to create and evaluate algorithms to predict credit risk. Six models will be created and compared to determine if any are worthy of recommendation for future use. The exercise will also demonstrate aspects of dealing with unbalanced classification.

## Results

We created and evaluated six algorithms:

1. Oversampling

2. SMOTE Oversampling

3. Undersampling

4. Combination (Over and Under) Sampling

5. Balanced Random Forest Classifier

6. Easy Ensemble AdaBoost Classifier

Precision is a measure of how likely a classification by the algorithm is accurate. The first observation, as expected, is that the precision score is meaningless for comparing the models as the data is significantly unbalanced. As such, Low\_risk received a score of 1.00 for all models. High\_risk received a score 0.01 to 0.07, with the Easy Ensemble AdaBoost Classifier model achieving the highest score. Solely based on this metric, it is difficult to tell the models apart or assess their strength other than to say Easy Ensemble AdaBoost Classifier did the best.

Balanced accuracy is a metric that one can use when evaluating how good a binary classifier is. It is especially useful when the classes are imbalanced, as in the case of credit risk, where good loans easily outnumber risky loans. The balanced accuracy scores were mediocre, although better than random for the first four models. Oversampling performed best of the four.

1. Oversampling – 0.65139

2. SMOTE Oversampling – 0.62669

3. Undersampling – 0.52927 (worst)

4. Combination (Over and Under) Sampling – 0.63108

When ensemble models were employed, balanced accuracy scores improved notably. Easy Ensemble AdaBoost Classifier’s score was the best of all six algorithms.

5. Balanced Random Forest Classifier – 0.79600

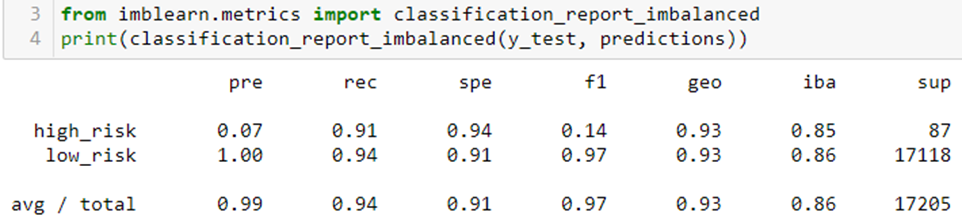
6. Easy Ensemble AdaBoost Classifier – 0.92525 (best)

Recall, or sensitivity, measures how many loans were correctly classified. Recall scores followed the path of improvement demonstrated by the balanced accuracy scores for the six models. For the first four, Undersampling had the worst recall scores, just as it had the worst balanced accuracy score. The ensemble models again performed the best, showing recall scores approaching 1.00. Furthermore, of all the models,

5. Balanced Random Forest Classifier – 0.91 (average)

6. Easy Ensemble AdaBoost Classifier – 0.94 (average)

The Easy Ensemble AdaBoost Classifier showed the best recall scores. Furthermore, of all the models, this model resulted in the best balance between the high\_risk (0.91) and low\_risk (0.94) scores, meaning the classifications were almost equally accurate for the two groups.



## Summary

Based on the metrics discussed above, one would feel justified recommending the Easy Ensemble AdaBoost Classifier be selected for future use. The obvious second choice would be the Balanced Random Forest Classifier model. When one reviews the respective confusion matrices, it becomes clearer that the ensemble models outperform the others with Easy Ensemble AdaBoost Classifier being the clear choice.

When looking at credit risk decisions, it helps to view the confusion matrix as below:

Table

Description automatically generated

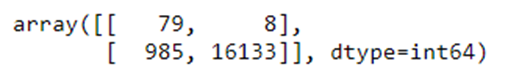
When classifying loans by risk, one wants to maximize the True Lows and True Highs. Further, one wants to minimize, if not eliminate False Lows. False Highs are undesirable, however, they are not as penalizing as False Lows. False Highs represent lost opportunities for Low\_risk loans that were incorrectly classified as High and therefore likely not to be made. Comparing the confusion matrices of the worst (Undersampling) to the best (Easy Ensemble AdaBoost Classifier) illustrates the range of difference in performance.

Undersampling

Graphical user interface, text

Description automatically generated

Easy Ensemble AdaBoost Classifier



Compared to all models, Easy Ensemble AdaBoost Classifier has maximized the True Lows and True Highs, and minimized False Lows and False Highs.

|  |  |  |
| --- | --- | --- |
|  | Predicted High | Predicted Low |
| Actual High | loans correctly not made | loans made to Hi |
| Actual Low | loans should have made | loans correctly made |

|  |  |  |
| --- | --- | --- |
|  | Predicted High | Predicted Low |
| Actual High | True High | False Low |
| Actual Low | False High | True Low |

|  |  |  |
| --- | --- | --- |
|  | Predicted High | Predicted Low |
| Actual High | Maximize | Minimize |
| Actual Low | Reduce | Maximize |

The goal is to maximize Predicted/Actual Low and minimize Predicted Low/Actual High.

balanced\_accuracy\_score = average of recall obtained on each class.

Precision = TP/(TP+FP)

Sensitivity = TP/(TP + FN)

F1 = 2(Precision \* Sensitivity)/(Precision + Sensitivity)

1. **Oversampling:**

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Table

Description automatically generated

1. **SMOTE Oversampling**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence



1. **Undersampling**

Graphical user interface, text

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. **Combination (Over and Under) Sampling**

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence



1. **Balanced Random Forest Classifier**

Graphical user interface, text, application

Description automatically generated



Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated



1. **Easy Ensemble AdaBoost Classifier**

Graphical user interface, text, application

Description automatically generated



A picture containing text

Description automatically generated



Graphical user interface, application

Description automatically generated

