MET CS 699 D1
DATA MINING AND BUSINESS INTELLIGENCE
SPRING 2016

PROJECT ASSIGNMENT

SUBMITTED BY,

RAVI KAILASAM RAJENDRAN
U30-05-9012
CS GRADUATE STUDENT
METROPOLITAN COLLEGE
BOSTON UNIVERSITY

SUBMITTED ON,

APRIL 14' 2016

Project assignment - Bank dataset

Since the given dataset for the analysis is an imbalanced dataset, the true positive rate of the models that are being created didn't give any convincing results. In order to increase the TP rate of the dataset, the oversampling of the yes class and the under sampling of the no class are being done and it gives better results.

Step1:

Initially the imbalanced data set is used to create four models with *cross validation of 10 folds* and it show the following results,

For NaiveBayes,

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.926	0.543	0.929	0.926	0.928	0.857	No
	0.457	0.074	0.444	0.457	0.45	0.857	Yes
Weighted Average	0.872	0.489	0.874	0.872	0.873	0.857	

Out of 3165 tuples, 2760 are correctly identified and most of them are No class.

The confusion matrix is given by

a	b	<—classified as
2594	208	a= no
197	166	b=yes

For J48 tree,

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.985	0.435	0.946	0.985	0.965	0.878	No
	0.565	0.015	0.83	0.565	0.672	0.878	Yes
Weighted Average	0.937	0.387	0.933	0.937	0.931	0.878	

Out of 3165 tuples, 2965 are correctly identified and most of them are No class.

The confusion matrix is given by

a	b	<—classified as
2760	42	a= no
158	205	b=yes

For Logistic regression,

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.979	0.705	0.915	0.979	0.946	0.895	No
	0.295	0.021	0.645	0.295	0.405	0.895	Yes
Weighted Average	0.9	0.627	0.884	0.9	0.884	0.895	

The confusion matrix is given by

a	b	<—classified as
2743	59	a= no
256	107	b=yes

For Neural Nets,

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.946	0.636	0.92	0.946	0.933	0.811	No
	0.364	0.054	0.466	0.364	0.409	0.811	Yes
Weighted Average	0.879	0.57	0.868	0.879	0.873	0.811	

The confusion matrix is

a	b	<—classified as
2651	151	a= no
231	132	b=yes

From this, we can tell that the TP rate of all the models are very low when the models are created using the imbalanced dataset.

Step 2:
The data is oversampled and the model is created for it. The oversampling is done using SMOTE (Synthetic Minority Oversampling TEchnique) algorithm.

Weka -> Filters -> Supervised -> Instance -> SMOTE

The percentage is set to 100 and it increases the yes tuples twice the current size and it is done repeatedly until the # of yes tuples matches the # of no tuples. So, the final dataset has 5706 tuples (2802 No and 2904 Yes) & randomized.
Naive Bayes,

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.805	0.66	0.922	0.805	0.859	0.929	No
	0.934	0.195	0.832	0.934	0.88	0.929	Yes
Weighted Average	0.87	0.132	0.876	0.87	0.87	0.929	

J48

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.958	0.046	0.952	0.958	0.955	0.98	No
	0.954	0.042	0.959	0.956	0.956	0.98	Yes
Weighted Average	0.956	0.044	0.956	0.956	0.956	0.98	

Logistic regression

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.898	0.071	0.925	0.898	0.911	0.957	No
	0.929	0.102	0.904	0.929	0.917	0.957	Yes
Weighted Average	0.914	0.086	0.914	0.914	0.914	0.957	

Neural Nets

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.981	0.02	0.979	0.981	0.98	0.984	No
	0.98	0.019	0.982	0.98	0.981	0.984	Yes
Weighted Average	0.98	0.2	0.98	0.98	0.98	0.984	

Step 3:
The data is under sampled and the model is created for it. The oversampling is done using SpreadSubSample in Weka

Weka -> Filters -> Supervised -> Instance -> SpreadSubsample

The distribution spread is set to 1 and it decreases the no tuples to the size of yes tuples. Now the dataset has 726 tuples (363 each) and they are randomized.
Naive Bayes,

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.788	0.198	0.799	0.788	0.793	0.864	No
	0.802	0.212	0.791	0.802	0.796	0.864	Yes
Weighted Average	0.795	0.205	0.795	0.795	0.795	0.864	

J48

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.934	0.074	0.926	0.934	0.93	0.971	No
	0.926	0.066	0.933	0.926	0.929	0.971	Yes
Weighted Average	0.93	0.07	0.93	0.93	0.93	0.971	

Logistic regression

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.857	0.171	0.834	0.857	0.845	0.913	No
	0.829	0.143	0.853	0.829	0.841	0.913	Yes
Weighted Average	0.843	0.157	0.843	0.843	0.843	0.913	

Neural Nets

	TP Rate	FP Rate	Precision	Recall	F-measure	ROC area	Class
	0.983	0.003	0.997	0.983	0.99	0.986	No
	0.997	0.017	0.984	0.997	0.99	0.986	Yes
Weighted Average	0.99	0.01	0.99	0.99	0.99	0.986	

The Neural networks performs better than other models in this dataset.

OTHER METHODS
This is also analyzed with the four models for different values of folds in the cross validation and using the percentage split of 66% and 80%.

The results of the models are given here

Cross validation with 5 folds
Naive Bayes for imbalanced initial dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.926     0.543      0.929     0.926     0.928      0.857    no
                 0.457     0.074      0.444     0.457     0.45       0.857    yes
Weighted Avg.    0.872     0.489      0.874     0.872     0.873      0.857

=== Confusion Matrix ===

    a    b   <-- classified as
 2594  208 |    a = no
  197  166 |    b = yes
```

Naive Bayes for over-sampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.804     0.303      0.953     0.804     0.873      0.823    no
                 0.697     0.196      0.316     0.697     0.435      0.823    yes
Weighted Avg.    0.792     0.291      0.88      0.792     0.822      0.823

=== Confusion Matrix ===

    a    b   <-- classified as
 2254  548 |    a = no
  110  253 |    b = yes
```

Naive Bayes for under-sampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.762     0.198      0.967     0.762     0.852      0.849    no
                 0.802     0.238      0.304     0.802     0.441      0.849    yes
Weighted Avg.    0.767     0.203      0.891     0.767     0.805      0.849

=== Confusion Matrix ===

    a    b   <-- classified as
 2135  667 |    a = no
   72  291 |    b = yes
```

J48 for imbalanced initial dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.985     0.435      0.946     0.985     0.965      0.878    no
                 0.565     0.015      0.83      0.565     0.672      0.878    yes
Weighted Avg.    0.937     0.387      0.933     0.937     0.931      0.878

=== Confusion Matrix ===

    a    b   <-- classified as
 2760   42 |    a = no
  158  205 |    b = yes
```

J48 for oversampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.958     0.289      0.962     0.958     0.96       0.933    no
                 0.711     0.042      0.686     0.711     0.698      0.933    yes
Weighted Avg.    0.93      0.261      0.931     0.93      0.93       0.933

=== Confusion Matrix ===

    a    b   <-- classified as
 2684  118 |    a = no
  105  258 |    b = yes
```

J48 for under sampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.814     0.074      0.988     0.814     0.893      0.905    no
                 0.926     0.186      0.392     0.926     0.551      0.905    yes
Weighted Avg.    0.827     0.087      0.92      0.827     0.854      0.905

=== Confusion Matrix ===

    a    b   <-- classified as
 2281  521 |    a = no
   27  336 |    b = yes
```

Logistic regression for imbalanced initial dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.979     0.705      0.915     0.979     0.946      0.895    no
                 0.295     0.021      0.645     0.295     0.405      0.895    yes
Weighted Avg.    0.9       0.627      0.884     0.9       0.884      0.895

=== Confusion Matrix ===

    a    b   <-- classified as
 2743   59 |    a = no
  256  107 |    b = yes
```

Logistic regression for oversampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.898     0.386      0.947     0.898     0.922      0.846    no
                 0.614     0.102      0.439     0.614     0.512      0.846    yes
Weighted Avg.    0.866     0.353      0.889     0.866     0.875      0.846

=== Confusion Matrix ===

    a    b   <-- classified as
 2517  285 |    a = no
  140  223 |    b = yes
```

Logistic regression for under-sampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.82      0.171      0.974     0.82      0.891      0.893    no
                 0.829     0.18       0.374     0.829     0.516      0.893    yes
Weighted Avg.    0.821     0.172      0.905     0.821     0.848      0.893

=== Confusion Matrix ===

    a    b   <-- classified as
 2299  503 |    a = no
   62  301 |    b = yes
```

Neural networks for oversampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.978     0.135      0.982     0.978     0.98       0.934    no
                 0.865     0.022      0.837     0.865     0.851      0.934    yes
Weighted Avg.    0.965     0.122      0.966     0.965     0.965      0.934

=== Confusion Matrix ===

    a    b   <-- classified as
 2741   61 |    a = no
   49  314 |    b = yes
```

Neural networks for under sampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.813     0.003      1         0.813     0.897      0.942    no
                 0.997     0.187      0.409     0.997     0.58       0.942    yes
Weighted Avg.    0.834     0.024      0.932     0.834     0.861      0.942

=== Confusion Matrix ===

    a    b   <-- classified as
 2279  523 |    a = no
    1  362 |    b = yes
```

Percentage split of data with 66%

Naive Bayes for imbalanced dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.926     0.543      0.929     0.926     0.928      0.857    no
                 0.457     0.074      0.444     0.457     0.45       0.857    yes
Weighted Avg.    0.872     0.489      0.874     0.872     0.873      0.857

=== Confusion Matrix ===

    a    b   <-- classified as
 2594  208 |    a = no
  197  166 |    b = yes
```

J-48 for imbalanced dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.985     0.435      0.946     0.985     0.965      0.878    no
                 0.565     0.015      0.83      0.565     0.672      0.878    yes
Weighted Avg.    0.937     0.387      0.933     0.937     0.931      0.878

=== Confusion Matrix ===

    a    b   <-- classified as
 2760   42 |    a = no
  158  205 |    b = yes
```

Logistic regression for imbalanced dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.979     0.705      0.915     0.979     0.946      0.895    no
                 0.295     0.021      0.645     0.295     0.405      0.895    yes
Weighted Avg.    0.9       0.627      0.884     0.9       0.884      0.895

=== Confusion Matrix ===

    a    b   <-- classified as
 2743   59 |    a = no
  256  107 |    b = yes
```

Neural Networks for imbalanced dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.996     0.118      0.985     0.996     0.99       0.949    no
                 0.882     0.004      0.964     0.882     0.921      0.949    yes
Weighted Avg.    0.983     0.105      0.982     0.983     0.982      0.949

=== Confusion Matrix ===

    a    b   <-- classified as
 2790   12 |    a = no
   43  320 |    b = yes
```

Percentage split of data with 80%

Checking the models with the imbalanced dataset using the percentage spit of 80 gives increased performance than the percentage split with 66%.

Hence applying it with the over and under sampled dataset, yields the following results.

Naive Bayes for under-sampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.797     0.237      0.753     0.797     0.775      0.873    no
                 0.763     0.203      0.806     0.763     0.784      0.873    yes
Weighted Avg.    0.779     0.219      0.781     0.779     0.779      0.873

=== Confusion Matrix ===

  a  b   <-- classified as
 55 14 |  a = no
 18 58 |  b = yes
```

J48 for under sampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.768     0.092      0.883     0.768     0.822      0.841    no
                 0.908     0.232      0.812     0.908     0.857      0.841    yes
Weighted Avg.    0.841     0.165      0.846     0.841     0.84       0.841

=== Confusion Matrix ===

  a  b   <-- classified as
 53 16 |  a = no
  7 69 |  b = yes
```

Logistic regression for under sampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.841     0.211      0.784     0.841     0.811      0.891    no
                 0.789     0.159      0.845     0.789     0.816      0.891    yes
Weighted Avg.    0.814     0.184      0.816     0.814     0.814      0.891

=== Confusion Matrix ===

  a  b   <-- classified as
 58 11 |  a = no
 16 60 |  b = yes
```

Neural Networks for under-sampled dataset

```
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.696     0.303      0.676     0.696     0.686      0.809    no
                 0.697     0.304      0.716     0.697     0.707      0.809    yes
Weighted Avg.    0.697     0.304      0.697     0.697     0.697      0.809

=== Confusion Matrix ===

  a  b   <-- classified as
 48 21 |  a = no
 23 53 |  b = yes
```

This is done for all the oversampled datasets and the results are obtained.

The above created four models in both the over and under sampling under Step 1 and Step 2 are again tested against a test data set created randomly from the initial dataset and they are split 60% for training , 20% for cross validation and the remaining 20 % for the testing with no tuples present in no more than one dataset.

The results with the models are
For Naive Bayes oversampled,

```
Correctly Classified Instances         510               80.5687 %
Incorrectly Classified Instances       123               19.4313 %
Kappa statistic                          0.3956
Mean absolute error                      0.2307
Root mean squared error                  0.374
Total Number of Instances              633

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.818     0.271      0.951     0.818     0.879      0.862    no
                 0.729     0.182      0.383     0.729     0.502      0.862    yes
Weighted Avg.    0.806     0.259      0.875     0.806     0.829      0.862

=== Confusion Matrix ===

   a   b   <-- classified as
 448 100 |   a = no
  23  62 |   b = yes
```

For Naive Bayes under-sampled,

```
Correctly Classified Instances         503               79.4629 %
Incorrectly Classified Instances       130               20.5371 %
Kappa statistic                          0.4099
Mean absolute error                      0.2765
Root mean squared error                  0.3917
Total Number of Instances              633

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.79      0.176      0.967     0.79      0.869      0.879    no
                 0.824     0.21       0.378     0.824     0.519      0.879    yes
Weighted Avg.    0.795     0.181      0.888     0.795     0.822      0.879

=== Confusion Matrix ===

   a   b   <-- classified as
 433 115 |   a = no
  15  70 |   b = yes
```

For J48 over-sampled

```
Correctly Classified Instances         590               93.207  %
Incorrectly Classified Instances        43                6.793  %
Kappa statistic                          0.6973
Mean absolute error                      0.097
Root mean squared error                  0.2409
Total Number of Instances              633

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.967     0.294      0.955     0.967     0.961      0.945    no
                 0.706     0.033      0.769     0.706     0.736      0.945    yes
Weighted Avg.    0.932     0.259      0.93      0.932     0.931      0.945

=== Confusion Matrix ===

   a   b   <-- classified as
 530  18 |   a = no
  25  60 |   b = yes
```

For J48 under-sampled

```
Correctly Classified Instances         538               84.9921 %
Incorrectly Classified Instances        95               15.0079 %
Kappa statistic                          0.5462
Mean absolute error                      0.1776
Root mean squared error                  0.3649
Total Number of Instances              633

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.836     0.059      0.989     0.836     0.906      0.927    no
                 0.941     0.164      0.471     0.941     0.627      0.927    yes
Weighted Avg.    0.85      0.073      0.92      0.85      0.869      0.927

=== Confusion Matrix ===

   a   b   <-- classified as
 458  90 |   a = no
   5  80 |   b = yes
```

For Logistic regression over-sampled

```
Correctly Classified Instances         550               86.8878 %
Incorrectly Classified Instances        83               13.1122 %
Kappa statistic                          0.48
Mean absolute error                      0.1796
Root mean squared error                  0.3185
Total Number of Instances              633

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.909     0.388      0.938     0.909     0.923      0.862    no
                 0.612     0.091      0.51      0.612     0.556      0.862    yes
Weighted Avg.    0.869     0.348      0.88      0.869     0.874      0.862

=== Confusion Matrix ===

   a   b   <-- classified as
 498  50 |   a = no
  33  52 |   b = yes
```

For Logistic regression under-sampled

```
Correctly Classified Instances         524               82.7804 %
Incorrectly Classified Instances       109               17.2196 %
Kappa statistic                          0.4682
Mean absolute error                      0.2453
Root mean squared error                  0.3595
Total Number of Instances              633

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.828     0.176      0.968     0.828     0.893      0.901    no
                 0.824     0.172      0.427     0.824     0.562      0.901    yes
Weighted Avg.    0.828     0.176      0.895     0.828     0.848      0.901

=== Confusion Matrix ===

   a   b   <-- classified as
 454  94 |   a = no
  15  70 |   b = yes
```

For Neural Nets over-sampled

```
Correctly Classified Instances         618               97.6303 %
Incorrectly Classified Instances        15                2.3697 %
Kappa statistic                          0.8976
Mean absolute error                      0.0302
Root mean squared error                  0.1533
Total Number of Instances              633

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.987     0.094      0.985     0.987     0.986      0.965    no
                 0.906     0.013      0.917     0.906     0.911      0.965    yes
Weighted Avg.    0.976     0.083      0.976     0.976     0.976      0.965

=== Confusion Matrix ===

   a   b   <-- classified as
 541   7 |   a = no
   8  77 |   b = yes
```

For Neural Nets under-sampled

```
Correctly Classified Instances         544               85.94   %
Incorrectly Classified Instances        89               14.06   %
Kappa statistic                          0.5807
Mean absolute error                      0.1473
Root mean squared error                  0.3498
Total Number of Instances              633

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.838     0          1         0.838     0.912      0.953    no
                 1         0.162      0.489     1         0.656      0.953    yes
Weighted Avg.    0.859     0.022      0.931     0.859     0.877      0.953

=== Confusion Matrix ===

   a   b   <-- classified as
 459  89 |   a = no
   0  85 |   b = yes
```

Step 4:

Observations and discussions:

The classifier models are analyzed using the test dataset and they are turned out to the following:

Doing under sampling always turns out to be the better model for predicting the yes tuple classes.

#Considering the models, each model discussed in the last section are tested against a common test dataset which is derived using the Resampling technique in the Weka

Weka -> Filters -> Unsupervised -> Instance -> Resample

The whole dataset (3165 tuples) is split into 60 % for training (1899 tuples which has 1702 No and 197 Yes) and then 20 % for cross validation(633 tuples) and then rest 20% for the testing dataset(633 tuples) with no set has a single tuple in common.

The best of all methods seems to be neural networks on the under sampled dataset as our goal is to predict the yes tuples in the imbalanced dataset correctly since being the minority.

Herewith, the attached files are the training dataset for the imbalanced dataset without doing sampling and then the model is the neural-network for under sampled data.