CSI 5155 Machine Learning
Homework Exercise 2
Wei Li 0300113733 report date: 11/1/2019

Part A

Question 1-3

The first half of Part A requires to rebalance the unbalanced Seismic Bumps data set and apply the four algorithms: a decision tree, a rule-based learning, a Naïve Bayesian classifiers and a k-nearest neighbor classifier to the three resampled data sets.

The data set has the following properties. Firstly, The data set is unbalanced, there are 2414 negative samples and only 170 positive samples. Then, There is no missing value. Also, four of the features in the data set is categorical, they are seismic, seismoacoustic, shift and ghazard. Since these features have orders, ordinal numbers start from zero were used to encode the these features.

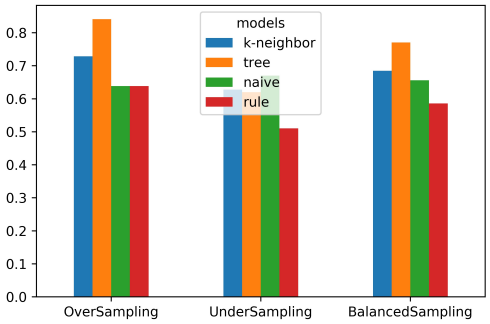
Ten fold cross-validation was used to evaluate the performance of classifiers using different sampled methods. In each iteration of the cross-validation, the data set was divided into a training set and a test set, and the three sampling method were used in the training set, while nothing was done to the test set. Then, the four classifiers were trained on these resampled training set, and were evaluated using the test set. In other word, since all the sampling methods were only applied to the training set, no bias from the sampling method could enter the test set, which was used to compare different sampling methods.

The three sampling method were chosen as follow:

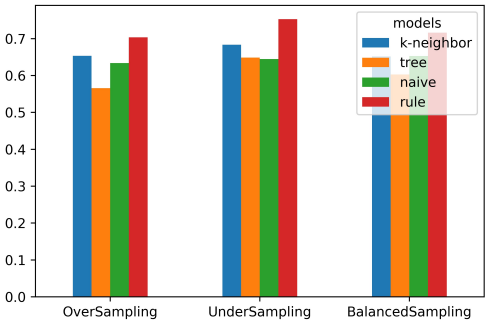
- Oversampling: SMOTE algorithm were used, the minority class(positive class) was over-sampled to the size of the majority class
- Under-sampling: random under-sampling were used, the majority class (negative class) was randomly drop to the size of the minority class
- Balanced sampling: we combined SMOTE oversampling and random under-sampling. Firstly, the majority class (negative class) was randomly drop to 5 times of the size of the minority class, and then the minority class was oversampled using SMOTE to the same size of the resampled majority class.

In each iteration, We used the test set to calculate three different metrics. They are: classifiers accuracy, recall and AUC. After the iterations, those metrics were averaged and then be used to compare different sampling method. The result can be plotted as Figure 1.

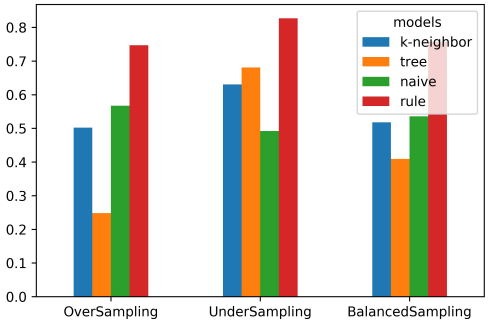
From the result, we can see that generally, under-sampling obtained the highest AUC and Recall, but obtained the worst accuracy. In other words, models trained using under-sampling tends to misclassify negative instance as positive, and thus have higher recall. In the seismic bumping case positive instance occurs in low possibility, so frequent False Positive prediction is not wanted. On the other hand, oversampling and balanced sampling got similar performance



(a) Accuracy of Different Sampling



(b) AUC of Different Sampling



(c) Recall of Different Sampling

Figure 1: Result comparison: the three different metrics score (averaged) obtained from different classifier trained using Oversampling, Under-sampling and Balanced sampling method.

in the three metrics, although individual classifier may vary. It can also be seen that for a classifier, there is a trade-off between recall and accuracy. Overall, balanced sampling got a more balanced result at accuracy and recall, so balanced sampling was chosen as the resampling method,

Question 4

The accuracies of the four classifier trained using balanced sampling method is shown in Table 1.

Table 2 shows the paired t-test result of the four models. Since no equal population variance is assumed here, Welch's t-test was performed. It can be seen that under 0.05 level of significance, we cannot reject the assumption that Naïve Bayes' performance is identical to the others, while the other three models are significantly different between each other.

	naïve	k neighbor	rule	tree
1	0.826254826	0.749035	0.6332046	0.833977
2	0.837837838	0.667954	0.6023166	0.76834
3	0.861003861	0.633205	0.5366795	0.749035
4	0.223938224	0.679537	0.5907336	0.752896
5	0.496124031	0.682171	0.5503876	0.825581
6	0.76744186	0.666667	0.5581395	0.736434
7	0.682170543	0.678295	0.5310078	0.771318
8	0.23255814	0.670543	0.6046512	0.736434
9	0.817829457	0.713178	0.7131783	0.79845
10	0.813953488	0.705426	0.5387597	0.732558
Avg	0.655911227	0.684601	0.5859058	0.770502
Std	0.249412907	0.031467	0.0565408	0.03709

Table 1: Models Accuracy Comparison

t-test	naïve-kneighbor	naïve-rule	naïve-tree
p-value	0.726253921	0.407137379	0.183141015
t-test	kneighbor-rule	kneighbor-tree	rule-tree
p-value	0.000265853	2.93E-05	2.55E-07

Table 2: T-test Results

Question 5-6

Question 5-6 requires to use two different feature selection techniques on the re-balanced data and retrain the four models to see if there is any difference in performance measured in accuracy. ANOVA based feature selection and Boruta method was used as the two feature selection technique.

Originally, the resampled data set has 18 features, and after applied the two feature selection methods to the data set separately, the number of features was reduced to 10. We did experiment on three different data sets, the first one was a dummy data set, i.e the original data set without feature selection, and the second one was the data set with Boruta feature selection, and the third was the data set with ANOVA feature selection. The four classifiers were trained and evaluated in the three data sets with 10-fold cross validation. The result is presented in Table 3-5.

The result shows that each classifier's generalization accuracy is similar with and without feature selection. We then did a paired t-test for each classifier and feature selection method, as shown in Table 6. The result shows that none of the feature selection method got a different accuracy comparing to the original data when the significance level is set to 0.05. As a result, feature selection did not improve the accuracy here.

#	naive	neigh	rule	tree
1	0.605882353	0.776470588	0.723529	0.782353
2	0.588235294	0.741176471	0.7	0.817647
3	0.552941176	0.770588235	0.717647	0.858824
4	0.623529412	0.747058824	0.7	0.841176
5	0.558823529	0.752941176	0.729412	0.829412
6	0.6	0.770588235	0.758824	0.835294
7	0.547058824	0.752941176	0.688235	0.782353
8	0.588235294	0.794117647	0.682353	0.829412
9	0.505882353	0.764705882	0.752941	0.794118
10	0.641176471	0.758823529	0.729412	0.841176
avg	0.581176471	0.762941176	0.718235	0.821176
std	0.040260589	0.015698525	0.025783	0.026481

Table 3: Boruta Feature Selection Result

#	naive	neigh	rule	tree
1	0.658824	0.729412	0.747059	0.847059
2	0.623529	0.741176	0.717647	0.776471
3	0.688235	0.788235	0.688235	0.852941
4	0.447059	0.782353	0.741176	0.852941
5	0.6	0.688235	0.664706	0.758824
6	0.670588	0.788235	0.723529	0.823529
7	0.647059	0.782353	0.723529	0.823529
8	0.641176	0.752941	0.711765	0.841176
9	0.641176	0.747059	0.776471	0.817647
10	0.629412	0.829412	0.682353	0.870588
avg	0.624706	0.762941	0.717647	0.826471
std	0.067115	0.039416	0.03316	0.035212

Table 4: ANOVA Feature Selection Result

#	naive	neigh	rule	tree
1	0.570588	0.758824	0.7	0.794118
2	0.611765	0.823529	0.729412	0.788235
3	0.505882	0.770588	0.688235	0.835294
4	0.605882	0.747059	0.729412	0.794118
5	0.558824	0.741176	0.652941	0.835294
6	0.529412	0.758824	0.688235	0.770588
7	0.635294	0.817647	0.729412	0.841176
8	0.676471	0.752941	0.770588	0.835294
9	0.594118	0.723529	0.717647	0.776471
10	0.564706	0.794118	0.735294	0.847059
avg	0.585294	0.768824	0.714118	0.811765
std	0.050316	0.032933	0.032716	0.029607

Table 5: Dummy Feature Selection Result

	naïve	neigh	rule	tree
dummy-Boruta	0.842241535	0.618761947	0.758373	0.463494
dummy-ANOVA	0.155980109	0.721584367	0.813351	0.325878
Boruta-ANOVA	0.09935353	1	0.965194	0.708745

Table 6: Paired t-test to Feature Selection

#	naive	neigh	rule	tree
1	0.911196911	0.938223938	0.942085	0.872587
2	0.918918919	0.922779923	0.92278	0.911197
3	0.926640927	0.930501931	0.934363	0.88417
4	0.891891892	0.911196911	0.895753	0.895753
5	0.887596899	0.937984496	0.930233	0.891473
6	0.926356589	0.949612403	0.934109	0.860465
7	0.918604651	0.903100775	0.914729	0.883721
8	0.910852713	0.941860465	0.937984	0.887597
9	0.868217054	0.906976744	0.903101	0.875969
10	0.88372093	0.926356589	0.903101	0.852713
avg	0.904399749	0.926859418	0.921824	0.881564
std	0.020146088	0.015744207	0.016609	0.017045

Table 7: Seismic Data Set Accuracy

Part B

Section requires to use the four models – decision tree, rule-based learning, Naïve Bayesian classifiers and k- nearest neighbor classifier – on three new data sets, and compare them with the generalization accuracy obtained from the Seismic Data Set. Those three data sets are: Labor Relations, Iris, and Congressional Voting.

- In the Labor Relations Data Set, there are only 57 instances and many missing values. Columns that have less than 30 values were dropped. Then mean impute were used to fill other missing values.
- In the Iris Data Set, there are 3 classes to be predicted. For classifiers that do not support multi-class classifying inherently, one-vs-all were used.
- In the Congressional Voting Data Set, voted for was denoted as 1, voted against was denoted as 0, votes that are neither positive nor negative was denoted as 0.5.

The four data sets were used on the four classifiers using 10-fold cross validation. The result is shown from Table 7 to Table 10.

Then, we placed all the average accuracy in a single table, Table 11, and did a Friedman's Test to see if the result obtained from different classifier, in different data set, are different.

The resulting Friedman statistic equals 5.153, while the p-value equals 0.161. Since the null hypothesis is that classifiers all have identical effects, and we cannot reject the null hypothesis under 0.05 significance level, it can be conclude that there is no significant difference in the accuracies obtained in question 2. Given the result, no post-hoc test is needed here.

#	naive	neigh	rule	tree
1	1	1	0.833333	0.833333
2	0.833333	1	0.5	0.666667
3	0.833333	0.833333	0.666667	0.833333
4	1	1	1	1
5	1	1	0.833333	0.666667
6	1	1	0.666667	0.833333
7	1	1	0.833333	0.833333
8	0.6	0.6	0.8	0.6
9	0.8	1	1	0.8
10	0.8	1	0.6	0.6
avg	0.886667	0.943333	0.773333	0.766667
std	0.136264	0.131515	0.163903	0.12862

Table 8: Labor Data Set Accuracy

#	naive	neigh	rule	tree
1	1	1	1	0.933333
2	1	1	1	1
3	0.933333	0.933333	0.933333	0.933333
4	0.933333	1	0.933333	0.933333
5	1	0.933333	1	1
6	1	1	0.933333	0.933333
7	0.933333	1	0.933333	1
8	0.933333	0.933333	0.933333	0.933333
9	1	1	1	1
10	0.8	0.866667	0.866667	0.866667
avg	0.953333	0.966667	0.953333	0.953333
std	0.063246	0.04714	0.044997	0.044997

Table 9: Iris Data Set Accuracy

#	naive	neigh	rule	tree
1	0.931818	0.909091	0.909091	0.954545
2	0.931818	0.931818	0.977273	0.931818
3	0.886364	0.909091	0.954545	0.909091
4	0.954545	0.954545	0.931818	0.954545
5	0.954545	0.909091	0.931818	0.909091
6	0.953488	0.930233	0.930233	0.930233
7	0.976744	0.930233	0.930233	0.953488
8	0.930233	0.930233	0.930233	0.930233
9	0.976744	0.976744	0.976744	0.976744
10	0.930233	0.953488	0.953488	0.906977
avg	0.942653	0.933457	0.942548	0.935677
std	0.026743	0.022447	0.02222	0.023635

Table 10: Vote Data Set Accuracy

Accuracy	naive	Kneighbor	rule	tree	Rank	naive	Kneighbor	rule	tree
Seismic	0.9044	0.9269	0.9218	0.8815	Seismic	3	1	2	4
Labor	0.8867	0.9433	0.7733	0.7666	Labor	2	1	3	4
Iris	0.9533	0.9667	0.9533	0.9533	Iris	2	1	2	2
Vote	0.9427	0.9335	0.9425	0.9356	Vote	1	4	2	3
Avg	0.9218	0.9426	0.8978	0.8843	Sum	8	7	9	13

Table 11: Average Accuracy and Rank