CSI 5155 Assignment2

Yu Sun(8472921)

October 2020

1 Question 1

1.1 Rebalance the data set

To balance the data set, I used the SMOTE for oversampling of the minority class, the NearMiss for under-sampling of the majority class and the SMOTEENN for balanced sampling combining oversampling and under-sampling. The sample size of both classes listed below.

Class	without rebalance	over-sampling	under-sampling	balanced sampling
Yes	4640	36548	4640	34849
No	36548	36548	4640	29674

Table 1: Sample Size with Different Method

1.2 Modeling

I used six model in total, including four models in assignment 1 and Random Forest as well as the Extreme Learning Tree.

About the KNN model($n\_neighbors$=3), the tenfold cross validation accuracy result with three sampling method listed in table 2.

About the Decision Tree model(max_depth=6 and $min\_samples\_leaf$=100), the tenfold cross validation accuracy result with three sampling method listed in table 3.

About the SVM model(C=20), the tenfold cross validation accuracy result with three sampling method listed in table 4.

About the Naive Bayesian model(GaussianNB), the tenfold cross validation accuracy result with three sampling method listed in table 5.

About the Random Forest model(max_depth=6 and $n\_estimators$=100), the tenfold cross validation accuracy result with three sampling method listed in table 6.

About the Extreme Learning Trees model(max_depth=6 and $n\_estimators$=100), the tenfold cross validation accuracy result with three sampling method listed in table 7.

	over-sampling	under-sampling	balanced sampling
Fold-1	0.9613	0.5453	0.9932
Fold-2	0.9632	0.667	0.9930
Fold-3	0.9553	0.68	0.9943
Fold-4	0.9525	0.597	0.9946
Fold-5	0.9491	0.625	0.9884
Fold-6	0.9337	0.6929	0.9950
Fold-7	0.8677	0.8136	0.9904
Fold-8	0.8392	0.8696	0.8122
Fold-9	0.8462	0.8696	0.9312
Fold-10	0.5938	0.8534	0.8326
Avg	0.8861	0.7213	**0.9524**

Table 2: KNN Tenfold Accuracy

	over-sampling	under-sampling	balanced sampling
Fold-1	0.7979	0.431	0.9278
Fold-2	0.9204	0.5043	0.9737
Fold-3	0.9304	0.6196	0.9772
Fold-4	0.9208	0.6153	0.9791
Fold-5	0.9223	0.4494	0.9440
Fold-6	0.8646	0.583	0.9758
Fold-7	0.6788	0.6724	0.9290
Fold-8	0.6009	0.8869	0.6885
Fold-9	0.7187	0.8513	0.8388
Fold-10	0.5934	0.7478	0.5288
Avg	0.7948	0.6360	**0.8762**

Table 3: Decision Tree Tenfold Accuracy

	over-sampling	under-sampling	balanced sampling
Fold-1	0.5073	0.6347	0.6918
Fold-2	0.8776	0.6735	0.9645
Fold-3	0.9917	0.6767	0.9967
Fold-4	0.9927	0.5916	0.9932
Fold-5	0.9918	0.5733	0.9919
Fold-6	0.9925	0.6024	0.9966
Fold-7	0.9914	0.8416	0.9974
Fold-8	0.6993	0.8998	0.9602
Fold-9	0.984	0.8438	0.8238
Fold-10	0.6209	0.8793	0.8489
Avg	0.8649	0.7216	**0.9264**

Table 4: SVC Tenfold Accuracy

	over-sampling	under-sampling	balanced sampling
Fold-1	0.6579	0.5582	0.7042
Fold-2	0.9347	0.5894	0.9540
Fold-3	0.9337	0.7155	0.9788
Fold-4	0.9447	0.6282	0.9802
Fold-5	0.9367	0.5194	0.9676
Fold-6	0.8379	0.5981	0.9706
Fold-7	0.8774	0.8675	0.9103
Fold-8	0.6673	0.8761	0.5538
Fold-9	0.8673	0.7608	0.7250
Fold-10	0.6694	0.8718	0.7804
Avg	0.8326	0.6984	**0.8524**

Table 5: Naive Bayesian Tenfold Accuracy

	over-sampling	under-sampling	balanced sampling
Fold-1	0.7462	0.4784	0.8536
Fold-2	0.9755	0.5409	0.9834
Fold-3	0.9435	0.6088	0.9848
Fold-4	0.9447	0.4461	0.9800
Fold-5	0.9409	0.4483	0.9651
Fold-6	0.9479	0.5539	0.9799
Fold-7	0.8539	0.7856	0.9743
Fold-8	0.6497	0.8793	0.5635
Fold-9	0.7076	0.8664	0.6604
Fold-10	0.5698	0.8459	0.5792
Avg	0.8279	0.6453	**0.8524**

Table 6: Random Forest Tenfold Accuracy

	over-sampling	under-sampling	balanced sampling
Fold-1	0.5387	0.4407	0.5983
Fold-2	0.9152	0.5334	0.9699
Fold-3	0.9179	0.5302	0.9936
Fold-4	0.9836	0.4978	0.9967
Fold-5	0.9818	0.4838	0.9856
Fold-6	0.9561	0.5291	0.9940
Fold-7	0.7602	0.6099	0.9868
Fold-8	0.6529	0.7489	0.5919
Fold-9	0.8967	0.7694	0.7832
Fold-10	0.6077	0.7662	0.8011
Avg	0.8210	0.5909	**0.8701**

Table 7: Extreme Learning Trees Tenfold Accuracy

For each model, I calculated the average accuracy of different sampling method, and I noticed that the balanced sampling could get the best result among them. So, I chose the balanced sampling method.

1.3 Evaluation

For the balanced sampling, the accuracy of six models list below.

	KNN	Decision Tree	SVM	Naive Bayesian	Random Forest	Extreme Learning Trees
Fold-1	0.9932	0.9278	0.6918	0.7042	0.8536	0.5983
Fold-2	0.9930	0.9737	0.9645	0.9540	0.9834	0.9699
Fold-3	0.9943	0.9772	0.9967	0.9788	0.9848	0.9936
Fold-4	0.9946	0.9791	0.9932	0.9802	0.9800	0.9967
Fold-5	0.9884	0.9440	0.9919	0.9676	0.9651	0.9856
Fold-6	0.9950	0.9758	0.9966	0.9706	0.9799	0.9940
Fold-7	0.9904	0.9290	0.9974	0.9103	0.9743	0.9868
Fold-8	0.8122	0.6885	0.9602	0.5538	0.5635	0.5919
Fold-9	0.9312	0.8388	0.8238	0.7250	0.6604	0.7832
Fold-10	0.8326	0.5288	0.8489	0.7804	0.5792	0.8011
Avg	0.9524	0.8762	0.9265	0.8525	0.8524	0.8701
Stdev	0.0677	0.1440	0.0989	0.1434	0.1702	0.1572

Table 8: The Accuracy of Six Models Using Balanced Sampling

Then, I calculated the pairwise differences in each fold between every two models and used paired t-test for statistical analysis. As we can see from the table below, KNN is statistically significant better than Decision Tree, Naive Beyesian and RF, while others did not have the statistical significance.

	KNN-DT	KNN-SVM	KNN-NB	KNN-RF	KNN-ELT	DT-SVM
Fold-1	0.0654	0.3014	0.2890	0.1396	0.3949	0.2360
Fold-2	0.0193	0.0285	0.0390	0.0096	0.0231	0.0092
Fold-3	0.0171	-0.0024	0.0155	0.0095	0.0007	-0.0195
Fold-4	0.0155	0.0014	0.0144	0.0146	-0.0021	-0.0141
Fold-5	0.0444	-0.0035	0.0208	0.0233	0.0028	-0.0479
Fold-6	0.0192	-0.0016	0.0244	0.0151	0.0010	-0.0208
Fold-7	0.0614	-0.007	0.0801	0.0161	0.0036	-0.0684
Fold-8	0.1237	-0.148	0.2584	0.2487	0.2203	-0.2717
Fold-9	0.0924	0.1074	0.2062	0.2708	0.1480	0.0150
Fold-10	0.3038	-0.0163	0.0522	0.2534	0.0315	-0.3201
Avg	0.0762	0.0260	0.1	0.1001	0.0824	-0.0502
Stdev	0.0833	0.1089	0.1024	0.1096	0.1265	0.1466
P-Value	**0.0226**	0.4923	**0.0168**	**0.0229**	0.0825	0.3310

	DT-NB	DT-RF	DT-ELT	SVM-NB	SVM-RF	SVM-ELT
Fold-1	0.2236	0.0742	0.3295	-0.0124	-0.1618	0.0935
Fold-2	0.0197	-0.0097	0.0038	0.0105	-0.0189	-0.0054
Fold-3	-0.0016	-0.0076	-0.0164	0.0179	0.0119	0.0031
Fold-4	-0.0011	-0.0009	-0.0176	0.0130	0.0132	-0.0035
Fold-5	-0.0236	-0.0211	-0.0416	0.0243	0.0268	0.0063
Fold-6	0.0052	-0.0041	-0.0182	0.0260	0.0167	0.0026
Fold-7	0.0187	-0.0453	-0.0578	0.0871	0.0231	0.0106
Fold-8	0.1347	0.1250	0.0966	0.4064	0.3967	0.3683
Fold-9	0.1138	0.1784	0.0556	0.0988	0.1634	0.0406
Fold-10	-0.2516	-0.0504	-0.2723	0.0685	0.2697	0.0478
Avg	0.0238	0.0238	0.0062	0.0740	0.0741	0.0564
Stdev	0.1183	0.0724	0.1417	0.1160	0.1518	0.108
P-Value	0.5615	0.3491	0.8991	0.0878	0.1771	0.1518

	NB-RF	NB-ELT	RF-ELT
Fold-1	-0.1494	0.1059	0.2553
Fold-2	-0.0294	-0.0159	0.0135
Fold-3	-0.006	-0.0148	-0.0088
Fold-4	0.0002	-0.0165	-0.0167
Fold-5	0.0025	-0.018	-0.0205
Fold-6	-0.0093	-0.0234	-0.0141
Fold-7	-0.064	-0.0765	-0.0125
Fold-8	-0.0097	-0.0381	-0.0284
Fold-9	0.0646	-0.0582	-0.1228
Fold-10	0.2012	-0.0207	-0.2219
Avg	0.0001	-0.0176	-0.0177
Stdev	0.0849	0.0457	0.1134
P-Value	0.9981	0.2768	0.6508

Table 9: The Result of Paired t-test

1.4 Feature Selection

For the feature selection, I chose two methods including removing features with low variance($threshold$=0.2) and model selection by Logistic Regression($penalty$=$l1$). The best two models I chose from the six models are KNN and SVM, which have high average accuracy and low standard deviation.

Through the paired t-test, we can see that for those two models, the p-values are not significant, which mean that the feature selection is useless for them. It is may because both models can get good results even without feature selection, the feature selection cannot make a big progress on them. If we could choose

	KNN-Based	KNN-Variance	KNN-L1	SVM-Based	SVM-Variance	SVM-L1
Fold-1	0.9932	0.9923	0.9930	0.6918	0.8886	0.8074
Fold-2	0.9930	0.9927	0.9929	0.9645	0.9495	0.9689
Fold-3	0.9943	0.9938	0.9944	0.9967	0.9630	0.9909
Fold-4	0.9946	0.9935	0.9941	0.9932	0.9650	0.9890
Fold-5	0.9884	0.9896	0.9887	0.9919	0.9670	0.9806
Fold-6	0.9950	0.9947	0.9950	0.9966	0.9670	0.9929
Fold-7	0.9904	0.9909	0.9902	0.9974	0.9735	0.9926
Fold-8	0.8122	0.8128	0.8126	0.9602	0.9423	0.9267
Fold-9	0.9312	0.9382	0.9349	0.8238	0.8452	0.8197
Fold-10	0.8326	0.8325	0.8348	0.8489	0.8131	0.8450
Avg	0.9524	0.9530	0.9530	0.9265	0.9274	0.9313
Stdev	0.0677	0.0673	0.0671	0.0989	0.0547	0.0731

Table 10: The Accuracy of before and after feature selection

some poor models, we can get that feature selection is useful. Besides, some models like SVM already have penalty and other regularization method, which can also reduce the space dimension of parameters. The other useful thing for feature selection is reduce the training time, because of the smaller parameter size.

	Variance-Based(KNN)	L1-Based(KNN)	Variance-Based(SVM)	L1-Based(SVM)
Fold-1	0.0009	0.0002	0.1968	0.1156
Fold-2	0.0003	0.0001	-0.015	0.0044
Fold-3	0.0005	-0.0001	-0.0337	-0.0058
Fold-4	0.0011	0.0005	-0.0282	-0.0042
Fold-5	-0.0012	-0.0003	-0.0249	-0.0113
Fold-6	0.0003	0.0	-0.0296	-0.0037
Fold-7	-0.0005	0.0002	-0.0239	-0.0048
Fold-8	-0.0006	-0.0004	-0.0179	-0.0335
Fold-9	-0.007	-0.0037	0.0214	-0.0041
Fold-10	0.0001	-0.0022	-0.035	-0.0039
Avg	0.0006	0.0005	0.0009	0.0048
Stdev	0.0022	0.0012	0.0670	0.0380
P-Value	0.4333	0.2078	0.9680	0.7102

Table 11: The Paired t-test of before and after feature selection

2 Question 2

I used three methods including SVM, k-NN and RF with the four data sets. The accuracy result listed below.

	Bank-KNN	Bank-SVM	Bank-RF	Labor-KNN	Laber-SVM	Laber-RF
Fold-1	0.9930	0.8074	0.8536	0.8333	0.6667	0.8333
Fold-2	0.9929	0.9689	0.9834	1.0	1.0	1.0
Fold-3	0.9944	0.9909	0.9848	1.0	1.0	1.0
Fold-4	0.9941	0.9890	0.9800	0.8333	0.8333	1.0
Fold-5	0.9887	0.9806	0.9651	1.0	0.8333	1.0
Fold-6	0.9950	0.9929	0.9799	1.0	0.8333	1.0
Fold-7	0.9902	0.9926	0.9743	0.8333	0.8333	1.0
Fold-8	0.8126	0.9267	0.5635	0.8	0.8	0.8
Fold-9	0.9349	0.8197	0.6604	0.8	1.0	0.8
Fold-10	0.8348	0.8450	0.5792	1.0	1.0	1.0
Avg	0.9530	0.9313	0.8524	0.91	0.88	0.9433
Stdev	0.0671	0.0731	0.1702	0.0907	0.1087	0.0869
	Voting-KNN	Voting-SVM	Voting-RF	Iris-KNN	Iris-SVM	Iris-RF
Fold-1	1.0	1.0	1.0	0.9773	0.9545	0.9773
Fold-2	1.0	0.9333	0.9333	0.9773	0.9091	0.9318
Fold-3	1.0	1.0	1.0	0.9545	0.9545	1.0
Fold-4	1.0	0.9333	0.9333	0.9545	0.8636	0.9545
Fold-5	0.9333	0.8667	0.9333	0.9773	0.9545	0.9773
Fold-6	1.0	1.0	0.9333	0.9535	0.907	0.907
Fold-7	0.8667	0.9333	0.9333	0.9767	0.9535	1.0
Fold-8	1.0	1.0	1.0	0.907	0.9767	0.9535
Fold-9	1.0	1.0	1.0	0.907	0.8372	0.907
Fold-10	1.0	1.0	1.0	0.907	0.907	0.9535
Avg	0.98	0.9666	0.9666	0.9492	0.9217	0.9561
Stdev	0.0426	0.0447	0.0333	0.029	0.0428	0.0319

Table 12: The Result of Three models

The I rank the average accuracy of each model for each data set, then used the Friedman Test to determine whether there is a statistically significant difference in the accuracy.

The Friedman statistic gives 2.5060 with n=4 and k=3, but the critical value is 7.8. So, we cannot reject the null hypothesis and all algorithms perform equally. Besides the critical difference is 1.6572 and the Nemenyi diagram showed in Figure 1.

Data set	KNN	SVM	RF
1	0.9530(1)	0.9313(2)	0.8524(3)
2	0.91(2)	0.88(3)	0.9433(1)
3	0.98(1)	0.9666(2)	0.9666(2)
4	0.9492(2)	0.9217(3)	0.9561(1)
Avg Rank	1.5	2.5	1.75

Table 13: The Rank of Three models

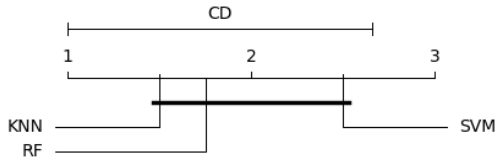

Figure 1: the Nemenyi diagram

3 Appendix

The code is available on GitHub:`https://github.com/YuSun09/CSI5155`