An Empirical Comparision to Supervised Learning Algorithms

Yushan Wang

[1]Department of Computer Science and Engineering, and Department of Mathematics
University of California, San Diego
* E-mail: yuw688@ucsd.edu

Several supervised learning methods have been learned and implemented in the lectures of *COGS 118A* . This document presents a ranking of four supervised learning methods: SVMs, Logistic Regression, Decision Tree, KNN, Gaussian Naive Bayes, bagging classifier. The experiment uses four data sets with different magnitudes as a variety of performance criteria to evaluate the four learning methods

1. Introduction

The past of the few weeks in the lectures of COGS 118A, I have been studying 2-class classification problem. The project is a rough reproduction to the *Empirical Comparison of Supervised Learning Algorithms* created by **Rich Caruana** and **Alexandru Niculescu-Mizil**. This project adopts 6 classifiers and 4 data sets with each has distinct size. This project presents the empirical comparison for six classifiers using two performance criteria. The project evaluates the performance of Logistic Regression, Decision Tree, SVM, KNN, Naive Bayes, bagging classifiers on four binary classification problems, using two performance metrics: accuracy and F-score. Moreover, a precious experience learned from the study this project is based on, For most of the

algorithms, we examine the variations and thoroughly explore the space of the parameters. For example, we compare, KNN with a great range of values of k, SVMs for different kernels, etc.

2. Methodology

2.1 Learning Algorithms

SVMs: We use the nonlinear kernels with different pairs of cost of mis-classification and parameter of a Gaussian Kernel. We pick five values from (10,-1) to (10,3) for **C** (cost of mis-classification) and **Gamma** (parameter of Gaussian Kernel) from list of [1e-6, 1e-5, 1e-4, 1e-3, 1e-2]

KNN: We use 8 of K ranging from K=1 to K =—train set— and choose the optimized one.

Bagging Classifier: We use the five classifiers: Logistic Regression, Decision Tree, KNN, Gaussian Naive Bayes, SVMs for the input for the estimators for the bagging.

2.2 Performance Metrics and Comparison

I mainly compare and evaluate threshold metrics for the performance metrics. Threshold metrics are accuracy and F-score.

There are two parameters which are restricted to binary cases: precision and recall pairs, and ROC(Compute Receiver operating characteristic). I use ROC for rank metrics, same as the methods from the empirical study.

Moreover, the problems in this projects, I are basically using two features to make prediction,

then another performance metrics would be based on 2-D visualization. To see the decision boundary region to have a clear classification result.

Comparison for performance Metrics, such as accuracy has value from 0 to 1 then I could compare the value for each classifier to determine the ranking for the classifiers.

Table 1: Descriptions of Data Sets

Problems	**Partition**	**Train Size**	**Test Size**	**Comments**
EX2	80/20	94	24	
EX2	50/50	59	59	
EX2	20/80	24	94	
BANKNOTE	80/20	1097	275	
BANKNOTE	50/50	686	686	
BANKNOTE	20/80	274	1098	
MAGIC	80/20	15216	3804	
MAGIC	50/50	9510	9510	
MAGIC	20/80	3804	15216	
SKIN	80/20	220552	24505	
SKIN	50/50	122528	122529	
SKIN	20/80	24505	220552	
SKIN	partition(5000)	5000	240057	This is a reproduction to the train size chosen to use in Rich Caruana's

2.3 Data Sets

I compare the six classifiers on 4 two-class classification problems. EX2 is from Kaggle, BANKNOTE and SKIN are from Machine Learning Repository of University of California, Irvine, and MAGIC is from Epistasis Lab at UPenn.

For the targeted class of MAGIC, I converted the letter form to number by setting letter ”g” to -1 and else to +1.

For the data preparation of data, except EX2, the rest of three data sets contains multi-features. I wished to do a 2-D features. Then I perform the PCA(Principal Component Analysis) to the three data sets. PCA is useful for eliminating dimensions, we took the first and second principal component values for columns for prediction.

For each data we would use three different partitions to divide the training sets and test sets. The partitions are set to : 20/80, 50/50, 80/20.

Check Table 1 in below to see the characteristics for the four problems.

3. Performance by metrics

To better reproduce the result to be as close as possible to the emprical study performed by the Rich Caruana and Alexandru Niculescu-Mizil, I performed similar selection to the data. Given that some of the questions do not have such a large data sets to satisfy the requirement to randomly select 100 points for training, so I performed the three different partitions on each data sets, use the numerator part for the training.

Table 2 shows the scores on each different parameters I choose to perform on Logistic Regression Classifier.

Table 2: Logistic Regression

KNN	Accuracy				F-score				ROC			
Param	**Set1**	**Set2**	**Set3**	**Set4**	**Set1**	**Set2**	**Set3**	**Set4**	**Set1**	**Set2**	**Set3**	**Set4**
lbfgs	**0.46**	**0.77**	**0.71**	**0.91**	**0.069**	**0.742**	0.356	**0.727**	**0.436**	**0.766**	**0.599**	0.845
newton-cg	0.46	0.77	0.71	0.91	0.069	0.742	0.356	0.727	0.436	0.766	0.599	0.845
sag	0.46	0.77	0.64	0.87	0.069	0.742	**0.471**	0.711	0.436	0.766	0.598	**0.923**
saga	0.46	0.77	0.63	0.80	0.069	0.742	0.464	0.645	0.436	0.766	0.590	0.881
liblinear	0.45	0.77	0.71	0.88	0.088	0.742	0.356	0.667	0.438	0.766	0.599	0.827

Overall, the Logistic Regression classifier performs quite poorly on all kernels, the difference between each is minute, and ”lbfgs” is slightly better than the others. Therefore, we would use Logistic Regression(random state=42, solver='lbfgs') for the performance by problems.

Table 3 shows the scores on each different parameters I choose to perform on the Support Vector Machine. The last row shows the one with the best parameters.

Table 3: SVM

Param	Accuracy				F-score				ROC			
Param	**Set1**	**Set2**	**Set3**	**Set4**	**Set1**	**Set2**	**Set3**	**Set4**	**Set1**	**Set2**	**Set3**	**Set4**
ovo	0.91	0.77	0.71	0.91	0.727	0.742	0.355	0.727	0.845	0.766	0.599	0.845
rbf	**1.00**	**1.00**	**1.00**	**1.00**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
linear	0.93	0.76	0.71	0.93	0.759	0.721	0.408	0.758	0.832	0.754	0.612	0.832

Table 4 shows the scores on each different k values which values of k are randomly selected. The last row shows the one with the best parameters.

The result of SVM and KNN is quite clear, kernel rbf and K=1 is clearly the best for each,

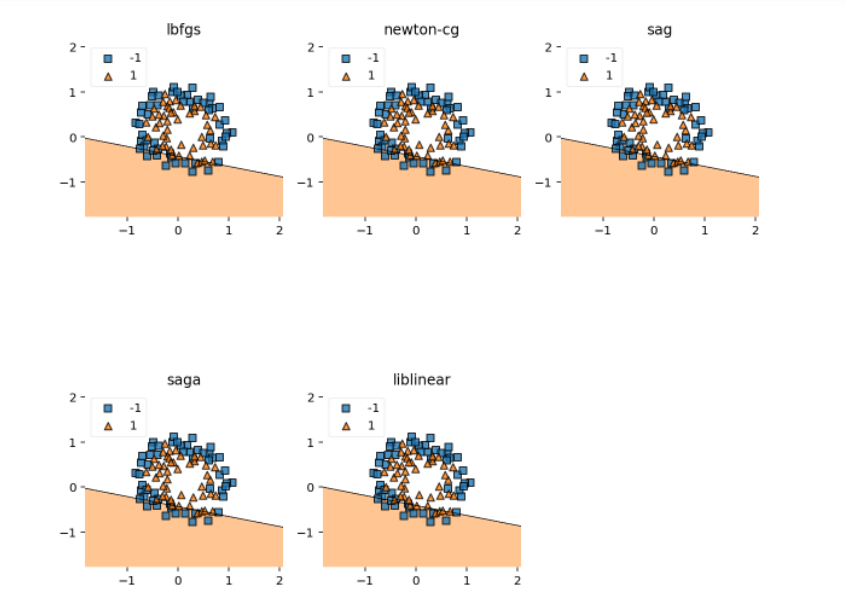

Figure 1: Visualization of Logistic Regression Classifier

but it still exists variations because the the metric sets' size is rather small.

4. Performance by Partitions on each problems

Now we are going to rank the six classifiers on each problems by taking mean from the three different partitions(training/ testing): 20/80, 50/50, 80/20. We calculate the accuracy, F-score, and ROC for the result of the prediction to the test data sets.

For each data sets has corresponding 3 different partitions and each partition has been repeated for three times and the results in the table are the average of results from each repeat.

The reproduction of the result of data 3 and 4 might be time consuming. Since the data sets are quite large. The specific number of the size of the data could be found in Table 1.

Table 4: KNN

Param	Accuracy				F-score				ROC			
Param	**Set1**	**Set2**	**Set3**	**Set4**	**Set1**	**Set2**	**Set3**	**Set4**	**Set1**	**Set2**	**Set3**	**Set4**
k=1	**1.00**	**1.00**	**1.00**	**1.00**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
k=3	0.88	0.97	0.79	0.97	0.862	0.921	0.687	0.914	0.880	0.927	0.759	0.982
k=7	0.84	0.87	0.74	0.97	0.826	0.843	0.458	0.914	0.838	0.862	0.642	0.982
k=15	0.83	0.83	0.72	0.93	0.825	0.813	0.364	0.787	0.831	0.828	0.607	0.882
k=29	0.69	0.78	0.67	0.83	0.716	0.738	0.108	0.261	0.698	0.772	0.529	0.570

Table 5: Problem : EX2

Param	Accuracy			F-score			ROC		
Param	**20/80**	**50/50**	**80/20**	**20/80**	**50/50**	**80/20**	**20/80**	**50/50**	**80/20**
LR	0.453	0.407	0.625	0.316	0.222	0.609	0.455	0.420	0.625
DT	0.568	0.729	0.792	0.594	0.758	0.815	0.568	0.725	0.792
GNB	0.600	0.627	0.625	0.457	0.56	0.609	0.603	0.637	0.625
SVM	0.653	**0.780**	**0.875**	0.535	**0.767**	**0.880**	0.655	**0.785**	**0.875**
KNN	**0.695**	0.644	0.708	**0.701**	0.656	0.741	0.695	0.644	0.708
Bagging	0.621	0.695	0.708	0.500	0.690	0.720	0.624	0.698	0.708

Problem EX2 has quite small data sets and the results could be found in table 5.

Results from the table shows the SVM would be the better classifier when the training sets getting large, and KNN shows the advantage when training size is small, and test size is large. The difference between each classifier is not large, except Logistic Regression. It would not be surprising when perform a linear classifier to a non-linear model.

Problem BANKNOTE has relatively large data sets which would be approximately ten times

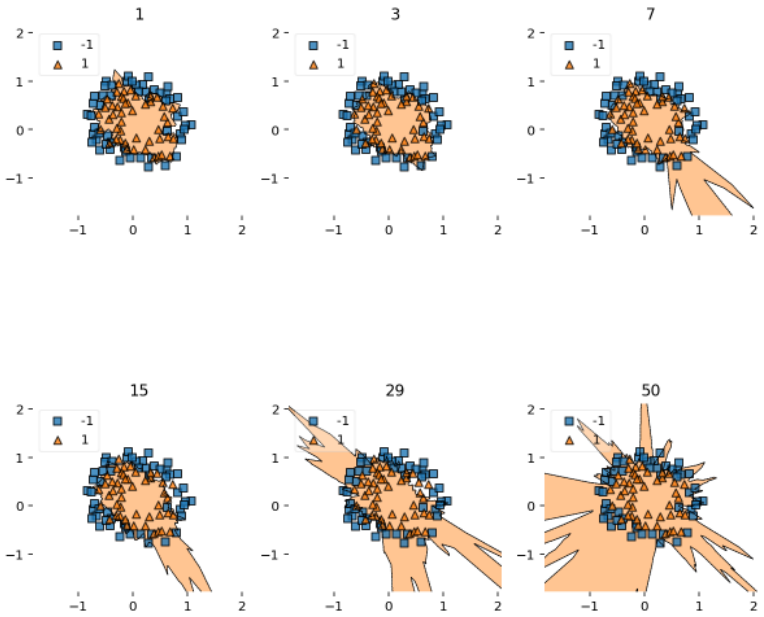

Figure 2: Visualization of KNN Classifier

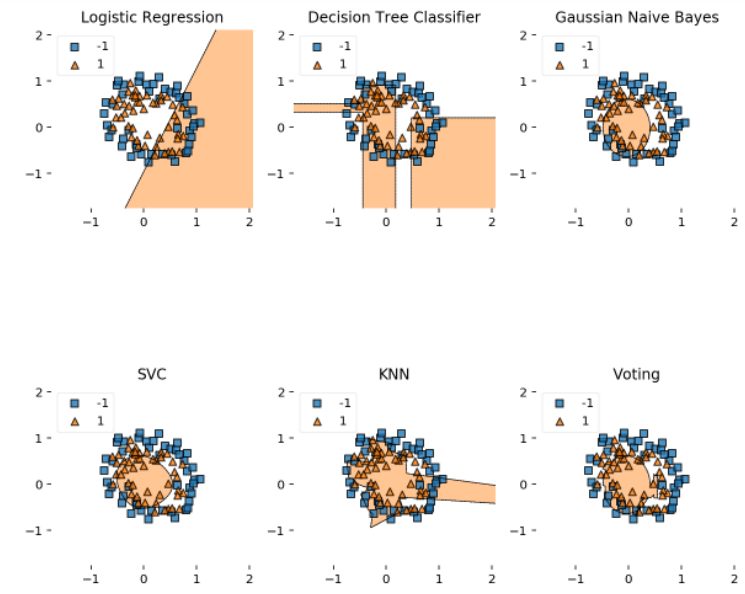

Figure 3: Visualization of EX2 on each classifier

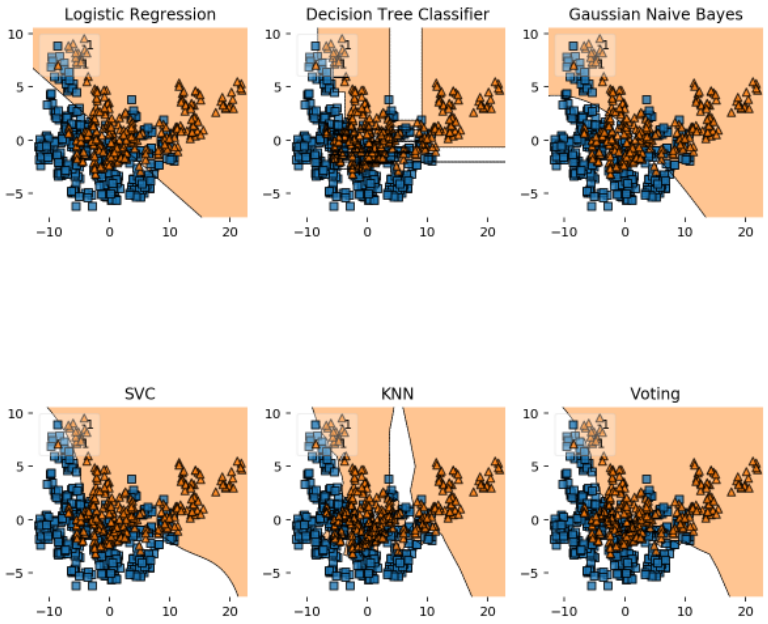

Figure 4: Visualization of BANKNOTE on each classifier

large as the EX2 data set. The results could be found in table 6.

Table 6: Problem : BANKNOTE

Param	Accuracy			F-score			ROC		
Param	**20/80**	**50/50**	**80/20**	**20/80**	**50/50**	**80/20**	**20/80**	**50/50**	**80/20**
LR	0.739	0.723	0.756	0.709	0.665	0.688	0.727	0.713	0.740
DT	0.807	0.806	0.836	0.788	0.771	0.796	0.807	0.801	0.826
GNB	0.716	0.701	0.724	0.664	0.624	0.635	0.707	0.686	0.703
SVM	0.801	0.816	0.833	0.780	0.783	0.808	0.801	0.811	0.834
KNN	**0.840**	**0.830**	**0.858**	**0.824**	**0.801**	**0.825**	**0.841**	**0.826**	**0.850**
Bagging	0.791	0.803	0.825	0.757	0.758	0.777	0.785	0.792	0.812

As again in the data banknote, the KNN stands out for the best, the decision tree and SVM approximately place at second place and LR and Gaussian perform poorly on this data set. The visualization is in below, Figure 4.

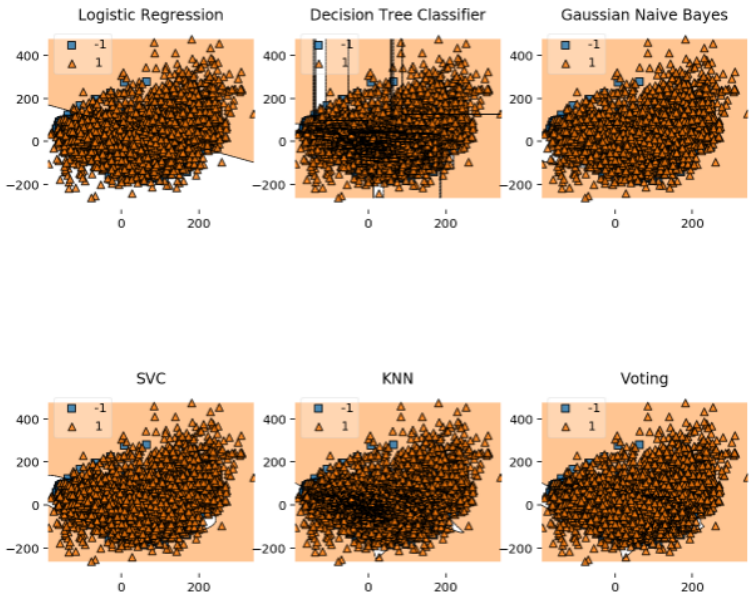

Figure 5: Visualization of MAGIC on each classifier

The data of MAGIC has largely overlap between label of +1 and -1. Therefore, the classification result on each classifier was quite bad performance, the result does not have the values to be shared. The visualization of both 2-D and 3-D could be found in Jupyter notebook. The results of each classifier performance, still be recorded in the notebook, but just does not have the potential of discussion in here. In the figure 6, The visualization would show the great overlap between labels.

Problem SKIN or NOT SKIN is a classic binary classification data sets. The size would be the largest which this project would be discussed. The size of this data is approximately 200 times large as the data of BANKNOTE.

The visualization in this problem is highly tricky to present two hundred thousands points in a

tiny sub-graph in the notebook, therefore, to testify the performance of each classifier on this data-set, I perform more than three partitions on this data set. The measures could be found in table 7.

Table 7: Problem : SKIN

Param	Accuracy			F-score			ROC		
Param	**20/80**	**50/50**	**80/20**	**20/80**	**50/50**	**80/20**	**20/80**	**50/50**	**80/20**
LR	0.912	0.912	0.912	0.792	0.790	0.790	0.873	0.871	0.871
DT	0.990	**0.992**	**0.993**	**0.976**	**0.981**	**0.984**	0.985	**0.989**	**0.990**
GNB	0.959	0.960	0.959	0.902	0.903	0.899	0.936	0.937	0.935
SVM	0.980	0.981	0.981	0.954	0.956	0.956	0.986	0.987	0.987
KNN	**0.990**	0.992	0.992	0.975	0.980	0.983	0.984	0.988	0.989
Bagging	0.985	0.986	0.986	0.965	0.967	0.968	**0.987**	0.988	0.989

Table 8: Problem : SKIN

Param	Accuracy		F-score		ROC	
Param	**90/10**	**5000(pt)/Rest**	**90/10**	**5000(pt)/Rest**	**90/10**	**5000(pt)/Rest**
LR	0.912	0.906	0.792	0.775	0.873	0.858
DT	**0.994**	0.982	**0.985**	0.956	**0.991**	0.973
GNB	0.957	0.959	0.897	0.900	0.933	0.932
SVM	0.980	**0.987**	0.954	**0.966**	0.986	**0.985**
KNN	0.993	0.984	0.984	0.961	0.990	0.976
Bagging	0.986	0.981	0.968	0.957	0.989	0.933

The performance on each partitions is quite outstanding. The size of training data sets are quite large, and decision tree and KNN 's performance over-weighted the others. However, there is an exception when partition is 5000/rest. The training set for the classifier is merely 5000 points versus two hundred thousands. The SVM shows the better performance than the others.

The accuracy for each classifier is not bad, but relatively, Gaussian Naive Bayes and Logistic regression perform quite poorly overall.

5. Cross validation

The method of implementing cross validation to each classifier is based on the similar task I performed on decision tree classifier on homework. The heat-map would demonstrate the result of error of each classifier corresponding to the corresponding parameters of the classifier. The table below shows each classifier's max depth and relatively optimized test error.

From previous results I acknowledged that the results would be better if the partition be set as 80/20. When the training data become sufficient, the classifier would be more accurate on the data. With less data to test, the error would be relatively smaller.Thus the cross validation is carried out on the partition of 80/20.

Firstly, we discuss the cross validation on decision tree. MaxDep stands for Max Depth and TestErr stands for test error.

Table 9: Cross Validation

Param	EX2		BANKNOTE		MAGIC		SKIN	
Param	**MaxDep**	**TestErr**	**MaxDep**	**TestErr**	**MaxDep**	**TestErr**	**MaxDep**	**TestErr**
DT	**4**	0.166	**3**	0.233	**5**	0.290	**5**	0.019

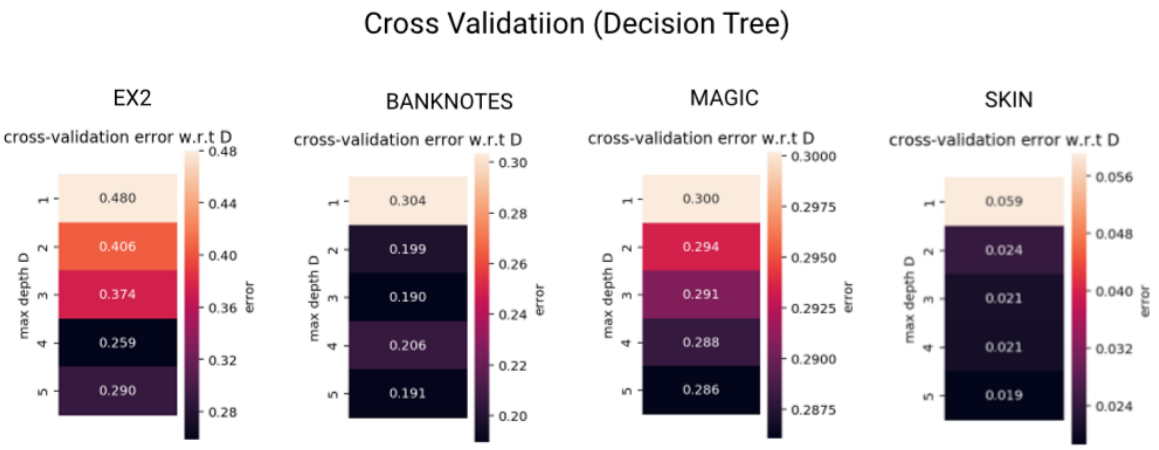

Figure 6: Heat Map of Decision Tree classifier

Logistic regression with the optimized C is what I am looking for when creating the heat-map and also to record the test error for the cross validation. The heatmap is in below, and result could be found in the table. The linear model of logistic regression is not highly suitable for the data sets I have which eventually results the test error are relatively large compared to other classifiers.

Table 10: Cross Validation

Param	EX2		BANKNOTE		MAGIC		SKIN	
Param	**Optc**	**TestErr**	**Optc**	**TestErr**	**Optc**	**TestErr**	**Optc**	**TestErr**
LR	**0.001**	0.458	**0.001**	0.233	**0.001**	0.427	**0.001**	0.124

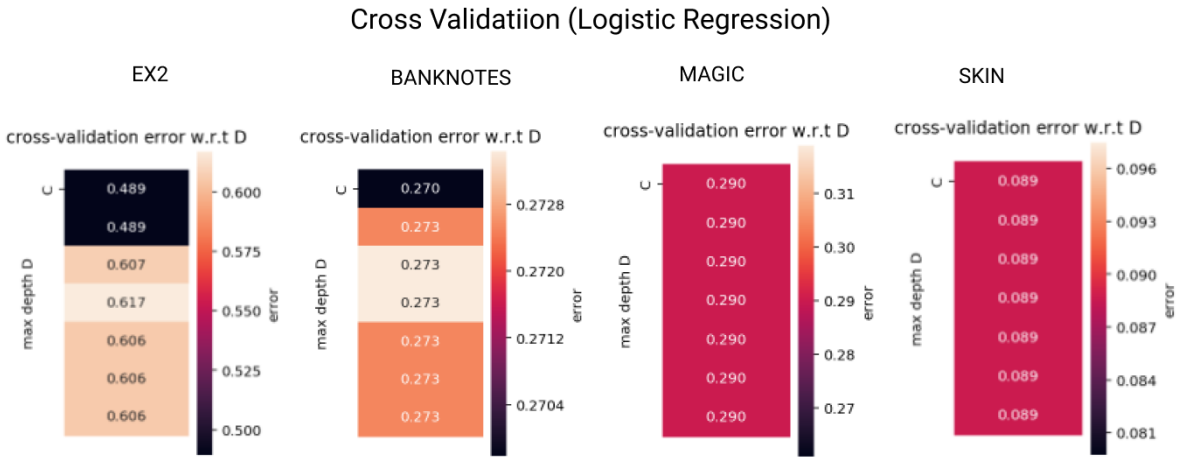

Figure 7: Heat Map of Logistic Regression classifier

For KNN, I set the parameters to be the different numbers of neighbors to get. Worth to mention, the data of Magic is quite unbalanced, then the result from that is biased.

Table 11: Cross Validation

Param	EX2		BANKNOTE		MAGIC		SKIN	
Param	**numN**	**TestErr**	**numN**	**TestErr**	**numN**	**TestErr**	**numN**	**TestErr**
LR	**5**	0.208	**3**	0.12	**47**	0.284	**1**	0.007

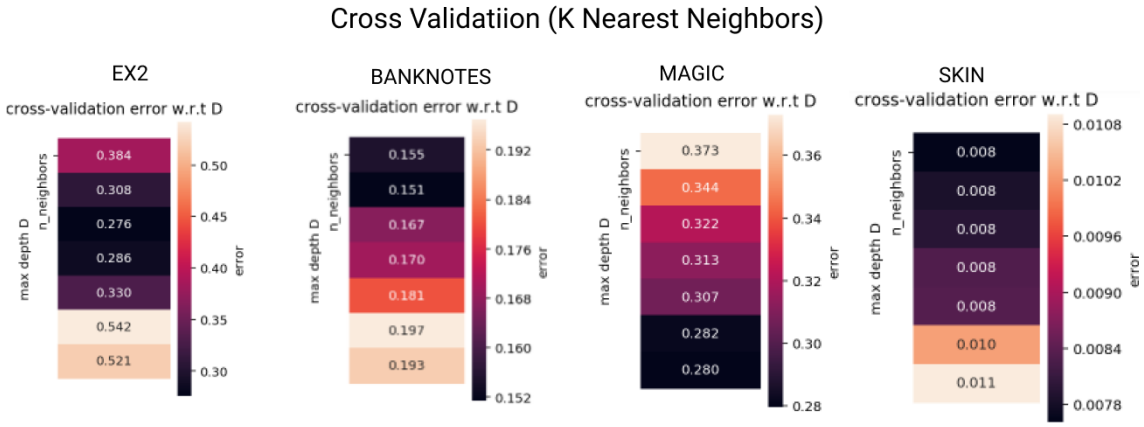

Figure 8: Heat Map of K-Nearest Neighbor classifier

6. Conclusion

Decision tree and KNN by using the sklearn module perform all quite well on each data-sets, even with overlapping data(ex: data3) and relative unbalanced data(ex: data 4). The bagging classifier and SVM classifier performed relatively well on each data sets. Moreover, the logistic regression classifier and Gaussian naive Bayes classifier are performed not so well.

The SVM classifier should have better results if more parameters could be experimented. However, the rbf kernel is the better one over the rest. It does slightly not match with the study results performed by Rich Caruana and Alexandru Niculescu-Mizil. The reason for this difference is that, the SVM the study applied to has calibration on the classifier. The calibration has many ways to apply to the classifier and greatly depend on the data sets. The calibration knowledge might beyond my current capability so far.

In a nutshell, the ranking of the classifiers basically match to the empirical study this project initially based on. Many performance metrics still could be applied to the classifiers. Exceptions

is also same occasions mentioned in the "An Empirical Comparison of Supervised Learning Algorithms". Even the best classifiers might performed quite poorly and some models with poor average performance occasionally perform exceptionally well.

7. Bonus

Per request, one individual project should perform 3 classifiers and 3 data sets. I choose 6 classifiers and 4 data sets for this project. The bagging classifier is the material beyond the lectures. Also, ROC(performance metrics) is also some part did not mention in the lectures but involved in the study, An Empirical Comparison of Supervised Learning Algorithms.

8. Acknowledgement

I thank Professor Zhuowen Tu for the advice of performing PCA(Principal Component Analysis) to prepare my data sets, Zihao Zhou for the SKIN data, for assistance received by using data from UCI Machine Learning Repository and for assistance received by using data from Epistasis Lab at UPenn.

9. Reference

Caruana, Rich. and Niculescu-Mizil,Alexandru. *An Empirical Comparison of Supervised Learning Algorithms* Liu, Bing. *Web Data Mining* **56**,(2007).

CieslakNitesh, David A. and Chawla, V. *Learning Decision Trees for Unbalanced Data* , **133**,2008

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science.

Moore, Jason H. research lab at the University of Pennsylvania, *A Python Automated Machine Learning tool that optimizes machine learning pipelines using genetic programming.*