Ziwen Zeng
A13456465
COGS118A

Final Project Report

Abstract:
Nowadays, machine learning algorithms come to be hot trends, which is incredibly powerful to make predications, classifications using large amounts of data. The comprehensive empirical evaluation of supervised machine learning has been popular since last century. Starting with neural networks and SVM (kernel) from 1970s, many new classification methods such as Random Forest and Bagging are standing out on stage. In the paper, we examine five classification methods: SVM(linear), K-Nearest Neighbors, Random Forests, Adaboost, Logistic Regression on four different datasets and compare how well they behave.

1. INTRODUCTION

This paper presents results of large-scale empirical comparison of five supervised machine learning algorithms. We evaluate the result the performance of SVM, Random Forest, K Nearest Neighbors, Adaboost and Logistic Regression using one performance metrics: Accuracy. The results are consistent with the result from *Caruana and Niculescu-Mizil*. Random Forest gives the best performance on three data sets. SVM, Adaboost perform well and K Nearest Neighbors and Logistic Regression perform relatively weaker.

2. METHOD

2.1 Learning Algorithms

We attempt to compare how well different classification method behave on same dataset and across different datasets. This section summarizes the parameter we used for each classification method. Each parameter list is based on the parameters from *Caruana and Niculescu-Mizil* and modified several times to make sure the best parameters we got from cross-validation are not on the edge of parameter list.

SVMs (SVM): We used the following kernels: linear, rbf. We vary the regularization parameter by factors from 10^{-3} to 10^{5} with each kernel.

Random Forest (RMF): The number of tree is considered 10, 20, 50, 80, 100.

K Nearest Neighbors (KNN): we use values of K neighbor ranging from K = 1 to K = 20.

Adaboost (ADB): The learning rate is considered 0.1, 0.2, 0.5, 0.8, 1. The number of trees is considered 20, 50, 80, 100.

Logistic Regression (LOG): We train by using regularization parameter from 10^{-3} to 10^{5}.

2.2 Performance Metrics

The threshold metric is accuracy (ACC). For each dataset, we calculated average train accuracy, average validation accuracy and average test accuracy over three trails for different classifier on different data partitions. The rank of classification method is based on the average test accuracy.

3. EXPERIMENT

3.1 Data Sets

We compare the algorithms on binary classification problems. We get four datasets: Wine Quality, Iris, Car Evaluation and Census Income from UCI online repository. We convert categorical data by one-hot encoding and we convert the predict label in to binary labels: +1 and -1.

Data_1 Wine quality: (data size: 1599, number of features: 12)
We normalize this dataset using the formula: (data – data.min)/(data.max-data.min). Then, we convert the predict label 'QUALITY' to binary label +1 when it's larger than 5 and -1 when it's less than 5.

Data_2 Iris: (data size: 149, number of features: 5)
We convert the predict label 'ClASS' to binary label +1 for 'Iris-setosa', -1 for 'Iris-virginica' and 'Iris-versicolor'.

Data_3 Car Evaluation: (data size: 1727, number of features: 17)
We first randomly sample 7000 data points out of 30,000 data points since original size is super large to run. We pick feature 'BUYING', 'MAINTENCE_PRICE', 'DOORS', 'PERSON', 'LUG_BOOT', 'SAFETY' and label 'ACCEPTABILITY' from data set. We convert BUYING, MAINTENCE_PRICE, LUG_BOOT and SAFETY by one-hot encoding. We convert the predict label 'ACCEPTABILITY' to binary label +1 for 'good, vgood', -1 for 'unacc', 'acc'.

Data_4 Census Income: (data size: 7000, number of features: 35)
We use data cleaning to remove 'NAN' data points. We pick feature 'AGE', 'WORKCLASS', 'CAPITAL GAIN', 'CAPITAL LOSS', 'EDUCATION', 'RACE', 'SEX' from dataset. We convert WORKCLASS, EDUCATION, RACE and SEX by one-hot encoding. We convert the predict label 'INCOME' to binary label +1 for income more than $50,000, -1 for less than or equal to $50,000.

We originally have five data sets but end up with four data sets: Data_1 (Wine quality), Data_3 (Iris), Data_4 (Car Evaluation), Data_5 (Census Income). One data set Data_2 was not examined in this experiment because the size of data set is too large to run.

3.2 Process

We first download four data sets from UCI online repository. Then, we start cleaning the data set by picking the features to use, dropping not available and messy data points, converting categorical feature by one-hot encoding and converting predict labels to binary labels. For each data set, we split data into three partitions (20% of all data for training and validation, 80% of all data for testing), (50% of all data for training and validation, 50% of all data for testing), (80% of all data for training and validation, 20% of all data for testing) and run four classifiers in three trails. For each classification method, we use three-fold cross-validation to find out the best hyper-parameters. (See Figure 5-8 for best hyper-parameters of four data sets). We report the best test accuracy (See Figure 1 - 4 for comprehensive graph of test accuracy over four data sets), train accuracy and validation accuracy (Figure 11 -18) under the chosen best hyper-parameter.

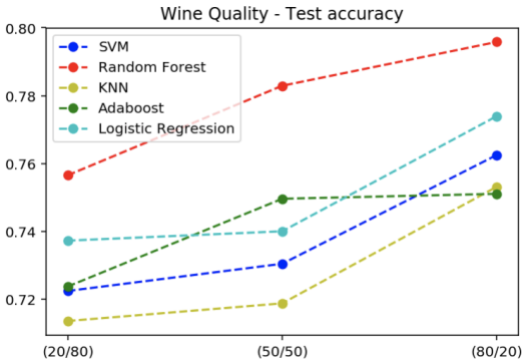

Figure 1

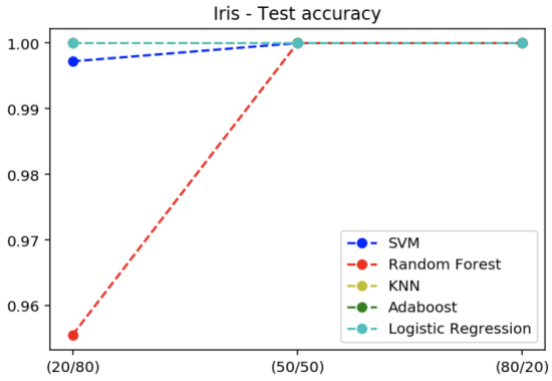

Figure 2

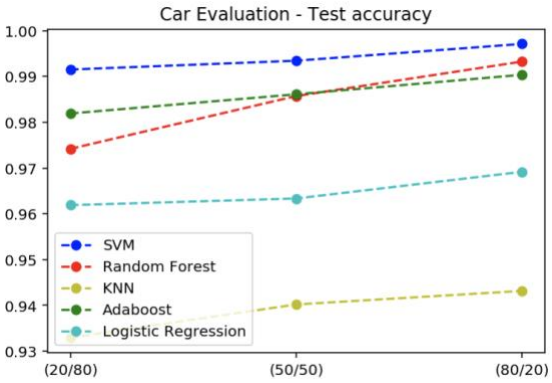

Figure 3

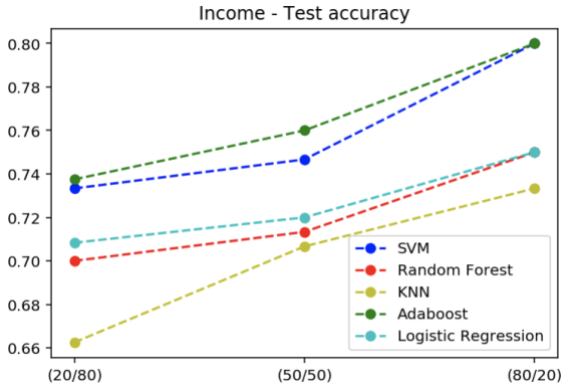

Figure 4

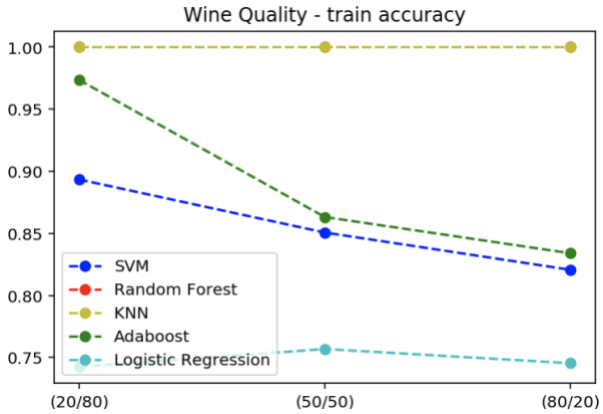

Figure 11

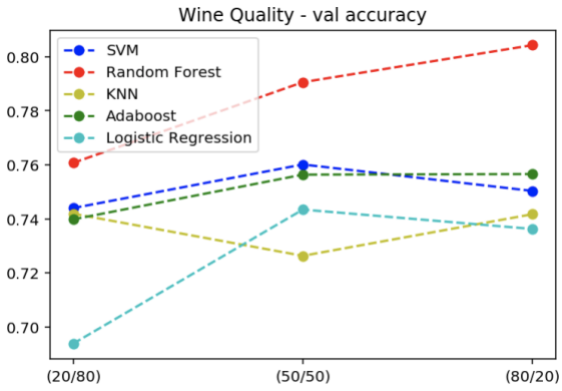

Figure 12

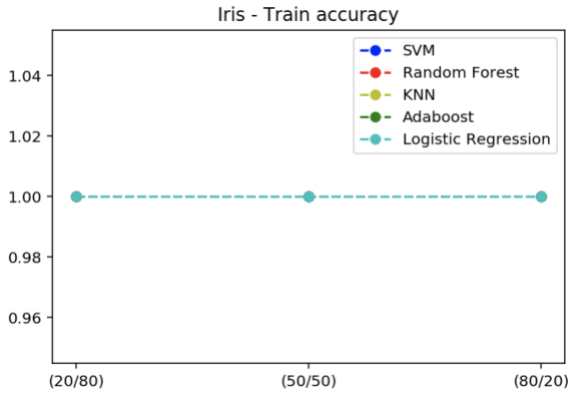

Figure 13

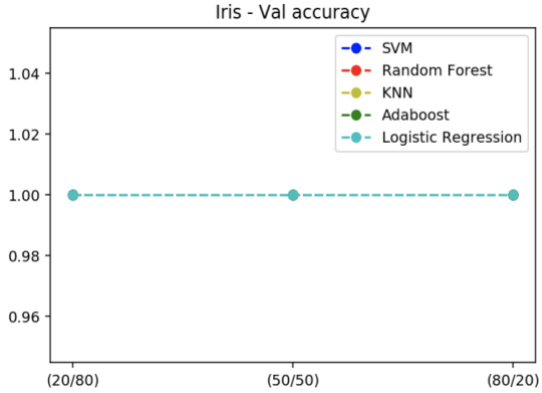

Figure 14

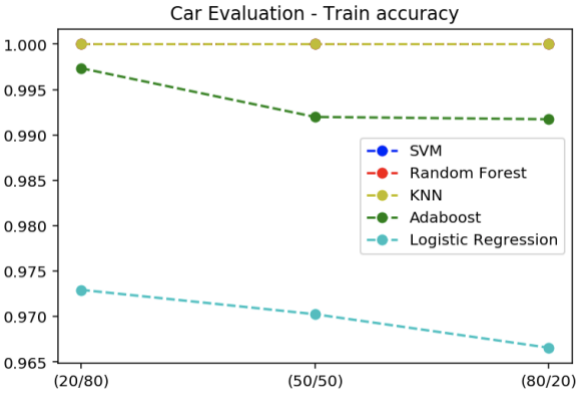

Figure 15

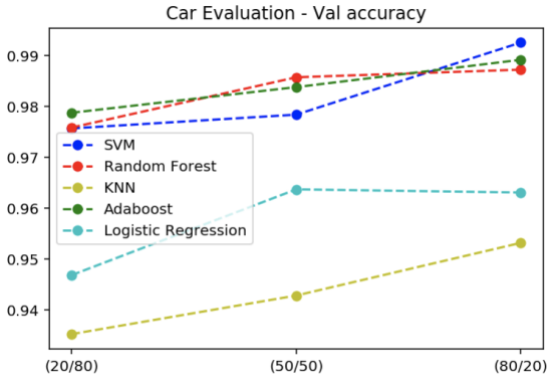

Figure 16

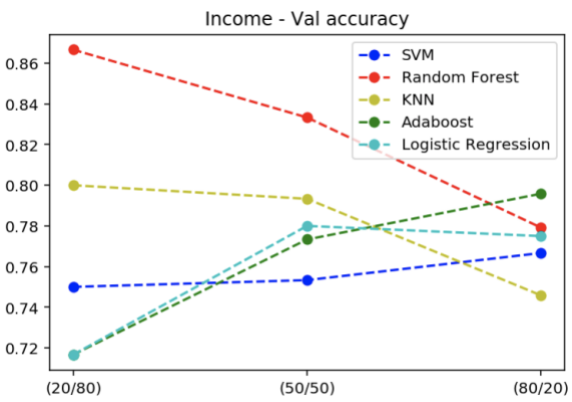

Figure 17

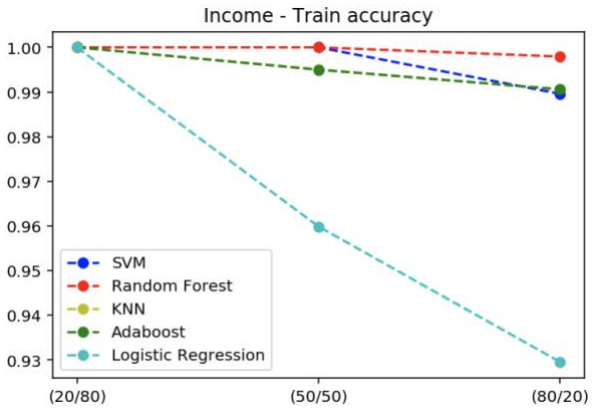

Figure 18

	Parameter	Partition		
		20/80	50/50	80/20
SVM	C	10000	1000	10000
	Kernal	rbf	linear	rbf
RMF	Criterion	gini	entropy	entropy
	Max feature	5	4	5
	# of Tree	100	50	50
KNN	K	15	12	2
ADB	Learning_rate	0.2	0.2	0.1
	# of Tree	50	80	50
LOG	C	10	10	10

Figure 5 (Best paramter for Data_1)

	Parameter	Partition		
		20/80	50/50	80/20
SVM	C	0.1	0.01	0.01
	Kernal	linear	linear	linear
RMF	Criterion	entropy	entropy	entropy
	Max feature	1	1	5
	# of Tree	20	10	10
KNN	K	1	1	1
ADB	Learning_rate	0.1	0.1	0.1
	# of Tree	20	20	20
LOG	C	0.1	0.1	0.1

Figure 6 (Best paramter for Data_3)

	Parameter	Partition		
		20/80	50/50	80/20
SVM	C	100	100	100
	Kernal	rbf	rbf	rbf
RMF	Criterion	entropy	entropy	entropy
	Max feature	7	7	14
	# of Tree	70	70	80
KNN	K	7	3	1
ADB	Learning_rate	0.5	0.8	0.5
	# of Tree	20	20	100
LOG	C	10	10	100

Figure 7 (Best paramter for Data_4)

	Parameter	Partition		
		20/80	50/50	80/20
SVM	C	0.001	0.001	0.001
	Kernal	linear	linear	linear
RMF	Criterion	entropy	entropy	gini
	Max feature	20	15	10
	# of Tree	20	20	10
KNN	K	1	8	12
ADB	Learning_rate	0.2	0.1	0.5
	# of Tree	20	20	50
LOG	C	10	0.01	0.01

Figure 8 (Best paramter for Data_5)

3.3 Coding

We use Python on Jupyter Notebook for code implementation of four classification methods. We imported SVC, RandomForestClassifier, KNeighborsClassifier, AdaboostClassifier and LogisticRegression functions from Scikit-learn, a software machine learning library which features various classification, regression and clustering alogrithms.

CONCLUSION

Below is the classification method ranking for four data sets:

Data_1: RMF has the highest test accuracy, ADB and SVM also has high test accuracy. KNN has the lowest test accuracy.

Data_3: This data set is very simple with only 149 data points and 4 features. All four classifiers work well with pretty high test accuracy. RMF, SVM and ADB has the highest test accuracy.

Data_4: SVM has the highest test accuracy, ADB and RMF also has pretty high test accuracy. KNN and LOG has the lowest test accuracy.

Data_5: ADB and SVM has the highest test accuracy. KNN has the lowest test accuracy.

The results are consistent with the result from *Caruana and Niculescu-Mizil*. Based on our experiment result (See figure 1- 4), the ascending lines for all four classifiers indicate that test accuracy increase when we use more training data. Overall, RMF and ADB works the best and SVM also works well with high accuracy. KNN and LOG work relatively weaker. With large training data (80/20), the test accuracy becomes high, approaching 0.95 ~ 1 for Iris and Car Evaluation data sets, 0.8 for Wine quality and Income data sets. Finally, we also compare the average test accuracy across four datasets on different classifiers and different partitions (See Figure 9). As Figure 10 indicates, RMF, SVM and ADB in general has the highest test accuracy over three partitions. Although we know that taking average test accuracy across different data sets might not be accurate, this is still part of indications.

	Classifier	20/80	50/50	80/20
0	SVM	0.854190	0.867629	0.889900
1	RMF	0.846560	0.870409	0.884772
2	KNN	0.827245	0.841387	0.857394
3	ADB	0.860775	0.873923	0.885325
4	LOG	0.851863	0.855835	0.873282

Figure 9: Average Test Accuracy Table across four data sets

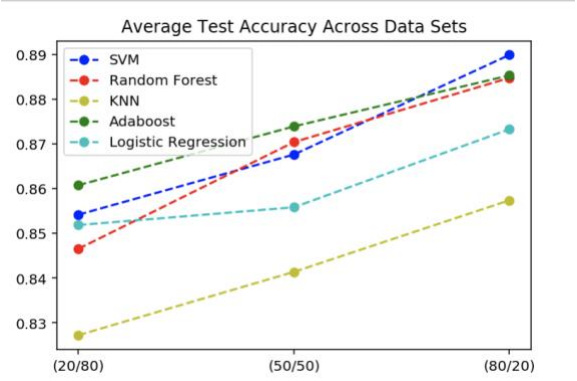

Figure 10: Average Test Accuracy Graph across four data sets

REFERENCES

Caruana Rich, Niculescu-Mizil Alexandru, *An Empirical Comparison of Supervised Learning Algorithms*
https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf

RESOURCES

Sklearn

BONUS POINTS:

1. I evaluated in total 4 data sets (Data_1, Data_3, Data_4, Data_5)
2. I used 5 classifiers (SVM, Random Forest, K Nearest Neighbors, Adaboost and Logistic Regression
3. My data sets are large: Data_1 with size 1599, Data_4 with size 1727 and Data_5 with size 7,000. This takes me very long time to run them all.
4. Comprehensive implementation: I used lots of visual representation (graphs) for data illustration.