An Empirical Comparison between Five Supervised Learning Methods

Table of Content

---

1. Abstract

I presented an empirical comparison between five supervised learning methods: Logistic Regression, SVMs, KNN, Decision Tree and Random Forest. By grid search, I chose the hyper-parameters for each classifier. Moreover, the comparison among five classifiers is based on the average test accuracy according to 3 partitions(20/80, 50/50, 80/20) * 3 datasets(Heart Disease, Bank Marketing and Breast Cancer) * 3 trails. The validation accuracy and train accuracy would be given in experiment part, but not in the conclusion part. After comparing the test accuracy, I believe SVM perform best among all five classifiers, although there is no great gap among those classifiers.

2. Introduction

I learned lots of supervised learning algorithms, linear regression, logistic regression, support vector machine, decision tree and so on. And this final project provides a great opportunity to practice comprehensively what I learned in class. We have multiple choices to analysis labeled data, and comparison among various supervised algorithms would built up a deeper understanding on each algorithm. This report presents results of empirical comparison of five supervised learning methods: Logistic Regression, Support Vector Machine(SVM), K Nearest Neighbors(KNN), Decision Tree and Random Forest.

3. Data and Problem Description

a. Dataset 1: Heart Disease Data Set

The data is related with heart disease. The classification goal is to predict if the patient will have a heart disease. The data contains 14 attributes including age, sex, chest pain type(cp), resting blood pressure(trestbps), serum cholesterol(chol), fasting blood sugar(fbs), resting electrocardiographic(restecg) and so on. And we have 303 instances. For more information on the dataset, please check this website: https://archive.ics.uci.edu/ml/datasets/Heart+Disease. The following picture showed the header of this dataset.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1

As we can see, the data looks good and the patient's id, ssn and name already been deleted before uploading. Since we only have 303 instances, instead of removing outliers, I normalized the given data. The normalized equation is:

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

b. Dataset 2: Bank Marketing Data Set

The data is related with direct marketing campaigns of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit. In this dataset, there are 21 attributes including age, type of job, marital status, education, housing loan, and so on. And we have 45211 instances. For more information on the dataset, please check this website: http://archive.ics.uci.edu/ml/datasets/Bank+Marketing#. The following picture showed the header of this dataset.

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome	emp.var.rate	co
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	...	1	999	0	nonexistent	1.1	
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	...	1	999	0	nonexistent	1.1	
2	37	services	married	high.school	no	yes	no	telephone	may	mon	...	1	999	0	nonexistent	1.1	
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	...	1	999	0	nonexistent	1.1	
4	56	services	married	high.school	no	no	yes	telephone	may	mon	...	1	999	0	nonexistent	1.1	

As we can see, there are lots of categorical data, so the next step is encode texts to numerical data. Next, I removed outliers by IQR method. Finally, I got clean dataset with sample size 40719, which is decent regarding to the original dataset. In this dataset, I predicted a binary variable - whether the client has subscribed a term deposit according to the left features.

Since when I did grid search in svm, my laptop still still run after 5 hours. I decided to make this dataset smaller. According to the data description, the features could be assigned to three labels: bank client data, last contract of the current campaign data, social and economic context attributes, and other attributes. In this data, we only focused on the social and economic context attributes to narrow down running time. After removing outliers by IQR and randomly choosing one percent of whole sample, the new dataset is 4119 * 5.

c. Dataset 3: Breast Cancer Data Set

The data is related with breast cancer. The classification goal is to predict the diagnosis is whether benign or malignant. The data contains 33 attributes including id, diagnosis(B for benign and M for malignant), radius_mean. texture_mean, perimeter_mean and so on. And we have 569 instances. For more information on the dataset, please check this website: https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original). The following picture showed the header of this dataset.

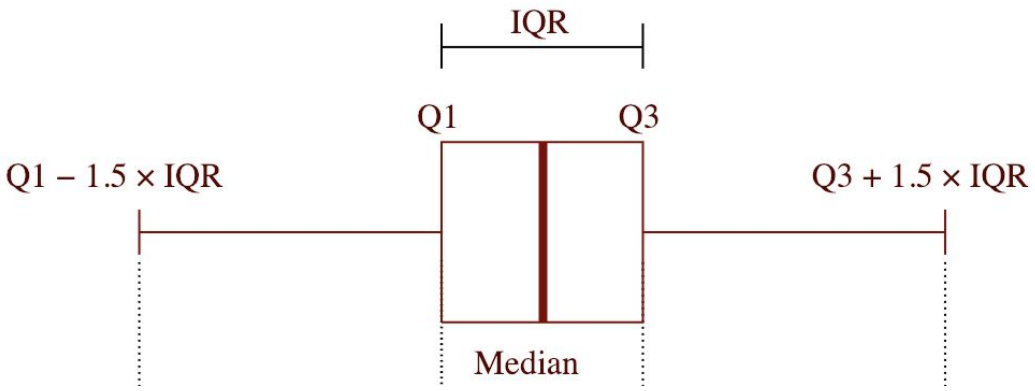
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	conc
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	

The first step is removing unrelated columns like “id”, “unnamed:32”, and I took “diagnosis” as target. Next, I removed outliers by IQR method. Finally, I got clean dataset with size 515 * 30, which is decent regarding to the original dataset. In this dataset, I predicted a binary variable - whether the the diagnosis is whether benign or malignant.

4. Method Description

a. Interquartile Range (IQR)

IQR is equal to the difference between 75th and 25th percentiles. Outliers defined as observations that fall below Q1 -1.5 IQR or above Q3+1.5 IQR. [1]



b. Logistic Regression[2]

I train both unregularized and regularized models, varying the ridge (regularation) parameter by factors of 10 from 10^{-8} to 10^{4}.

c. Support Vector Machines (SVMs)[3]

I use the following kernels in SVMLight (Joachims, 1999): linear, polynomial degree 2 & 3, radial with width {0.001,0.005,0.01,0.05,0.1,0.5,1,2}. I also vary the regularization parameter by factors of ten from 10^{-7} to 10^{3} with each kernel.

---

[1] https://en.wikipedia.org/wiki/Interquartile_range

[2] https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf

[3] https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf

d. K-Nearest Neighbors (KNN)[4]

I use 26 values of K ranging from K = 1 to K = 26. I use KNN with Euclidean distance and Euclidean distance weighted by gain ratio. I also use distance weighted KNN, and locally weighted averaging. The kernel widths for locally weighted averaging vary from 2^0 to 2^{10} times the minimum distance between any two points in the train set.

e. Bonus: Decision Tree[5]

I trained decision tree with max depth from 1,2,4,6,8,12,16, 20 with criterion entropy for the information gain.

f. Bonus:Random Forests (RF)[6]

The forests have 1024 trees. The size of the feature set considered at each max_depth is 1,2,4,6,8,12,16 or 20.

g. Metric: Classification Accuracy

I used accuracy matrix to compare the performance for each classifier. The accuracy output is between 0 to 1, the higher test accuracy means the model is better.

[4] https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf

[5] https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf

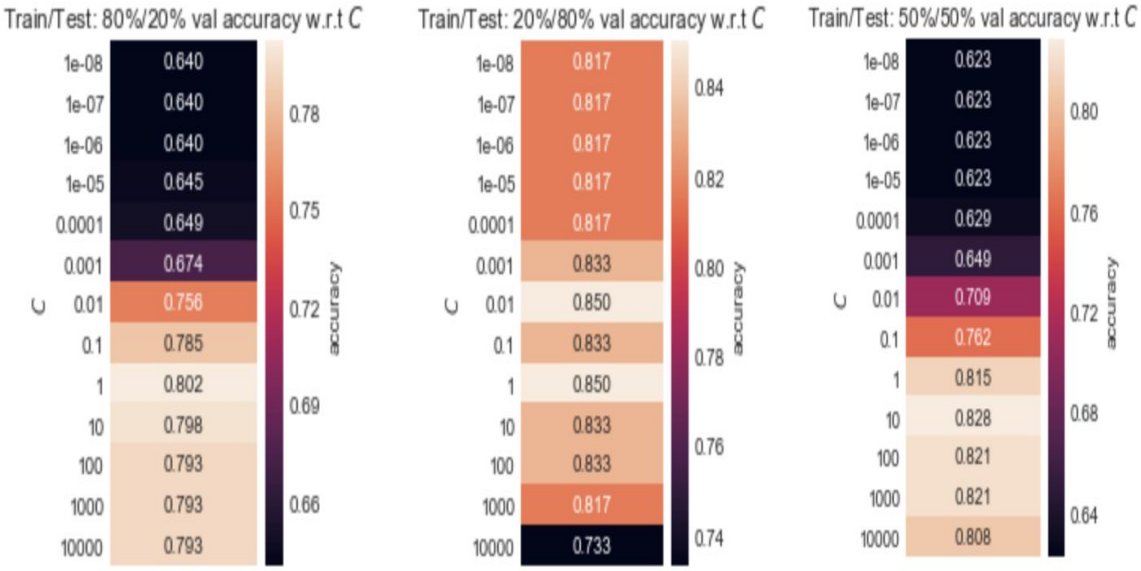
[6] https://www.cs.cornell.edu/~caruana/ctp/ct.papers/caruana.icml06.pdf

5. Experiments

I divided Experiments to three parts with regard to three datasets: Heart Disease, Bank Marketing and Breast Cancer.

a. Heart Disease Dataset

i. Logistic Regression



This is three heatmaps of validation accuracies with varying the ridge (regularation) parameter from 10^{-8} to 10^{4} for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train acc is: 0.84, test accuracy is: 0.85 with C = 1.00.

For 20% training and 80% testing, train acc is: 0.85, test accuracy is: 0.79 with C = 0.01.

For 50% training and 50% testing, train acc is: 0.89, test accuracy is: 0.83 with C = 10.00.

ii. SVMs

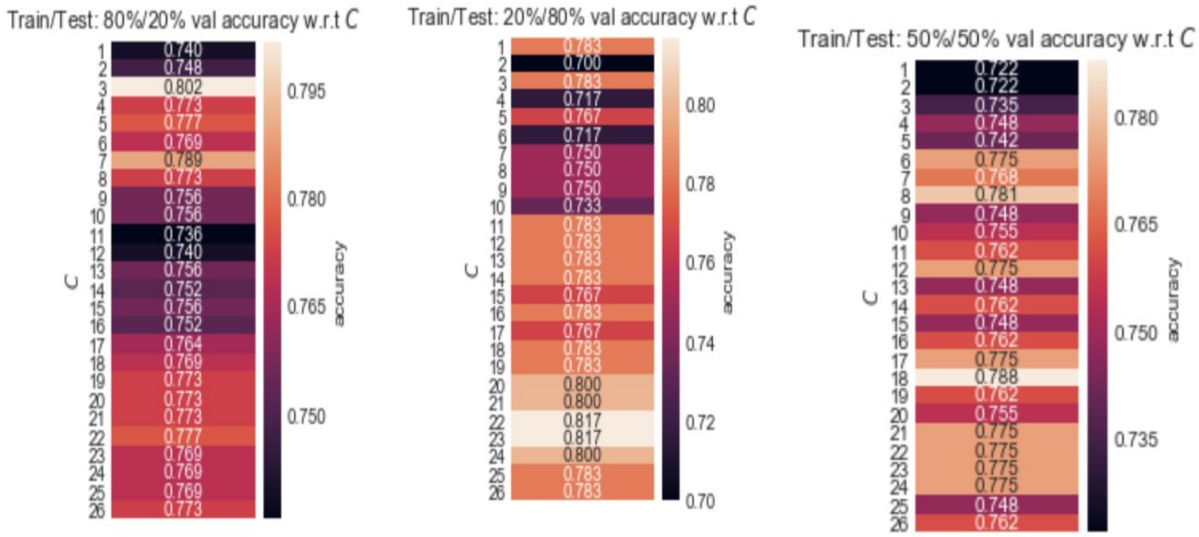
In SVM part, since I changed three parameters: degree, gamma, C in SVM classifier, it is hard to show validation accuracy based on the plot or table. Then I just gave the model and accuracies with optimal parameter.

For 80% training and 20% testing, train accuracy is: 0.86, test accuracy is: 0.89 and the best parameter is degree = 3, radial width =2.00, regularization parameter=0.001.

For 20% training and 80% testing, train accuracy is: 0.88, test accuracy is: 0.76 and the best parameter is degree = 3, radial width =2.00, regularization parameter=0.001.

For 50% training and 50% testing, train accuracy is: 0.89, test accuracy is: 0.79 and the best parameter is degree = 3, radial width =2.00, regularization parameter=0.001.

iii. KNN



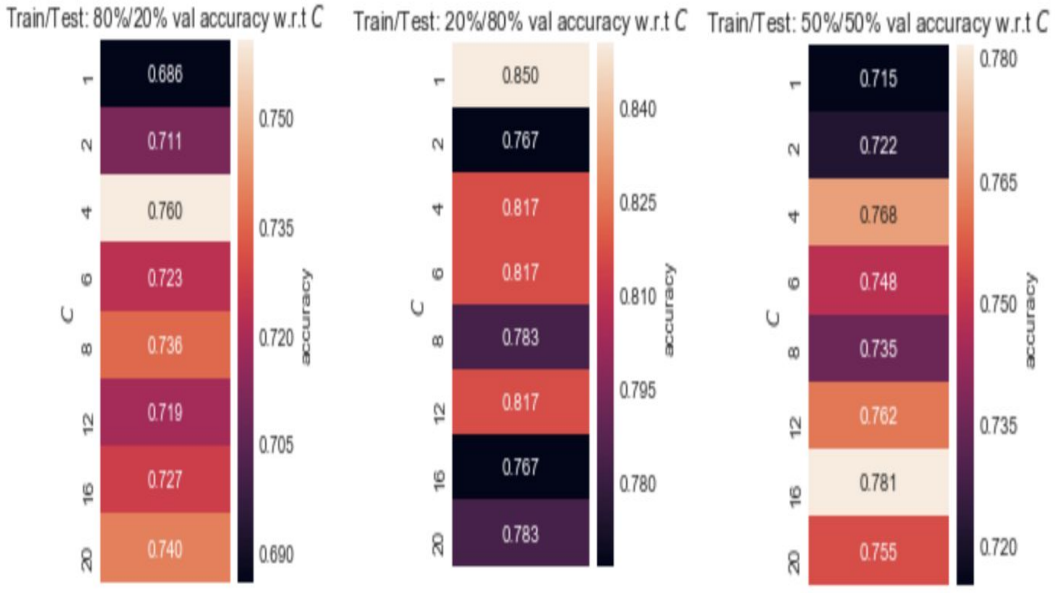
The above of validation accuracies with varying k from 1 to 26 for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train accuracy is: 0.89, test accuracy is: 0.84 with k = 3.00.

For 20% training and 80% testing, train accuracy is: 0.78, test accuracy is: 0.79 with k = 22.00.

For 50% training and 50% testing, train accuracy is: 0.80, test accuracy is: 0.80 with k = 18.00

iv. Decision Tree



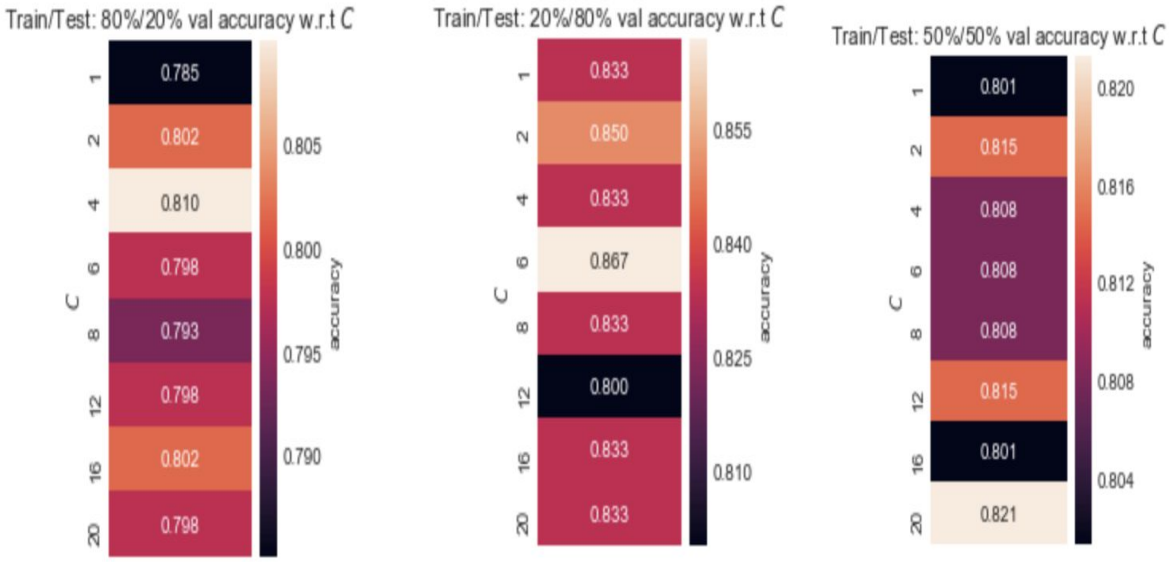
The above of validation accuracies with varying max_depth 1,2,4,6,8,12,16 or 20 for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train accuracy is 0.89 and test accuracy is: 0.89 with max_depth = 4.

For 20% training and 80% testing, train accuracy is 0.89 and test accuracy is: 0.74 with max_depth = 1.

For 50% training and 50% testing, train accuracy is 1.00 and test accuracy is: 0.80 with max_depth = 16.

v. Random Forest



The above of validation accuracies with varying max_depth 1,2,4,6,8,12,16 or 20 for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train accuracy is: 0.92 and test accuracy is: 0.87 with max_depth = 4.

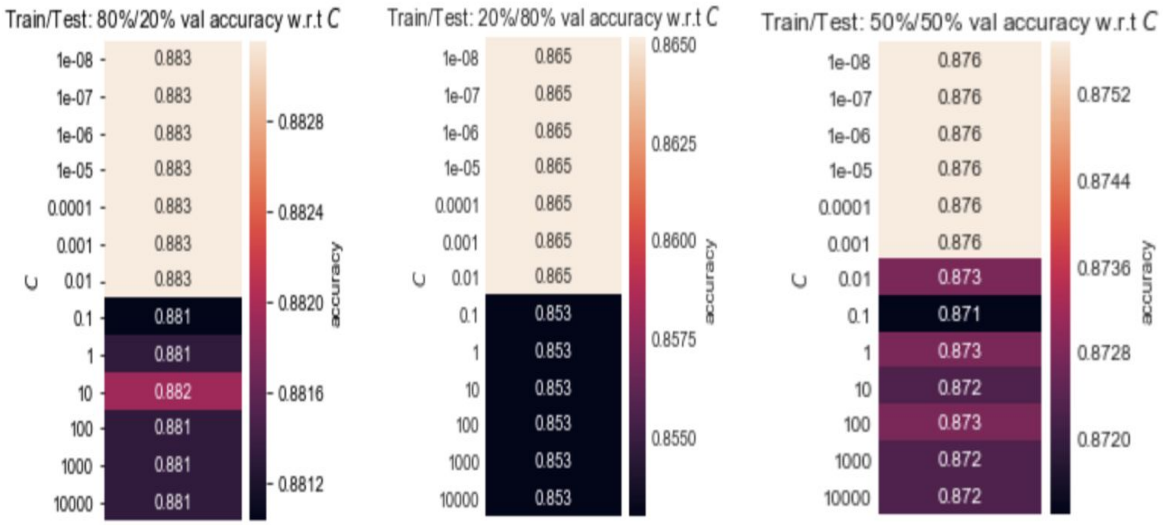
For 20% training and 80% testing, train accuracy is: 1.00 and test accuracy is: 0.81 with max_depth = 6.

For 50% training and 50% testing, train accuracy is: 1.00 and test accuracy is: 0.84 with max_depth = 20.

.

b. Bank Marketing Dataset

i. Logistic Regression



This is three heatmaps of validation accuracies with varying the ridge (regularation) parameter from 10^{-8} to 10^{4} for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train acc is: 0.88, test acc is: 0.89 with C = 0.00.

For 20% training and 80% testing, valtrain acc is: 0.87, test acc is: 0.89 with C = 0.00.

For 50% training and 50% testing, train acc is: 0.88, test acc is: 0.89 with C = 0.00.

ii. SVMs

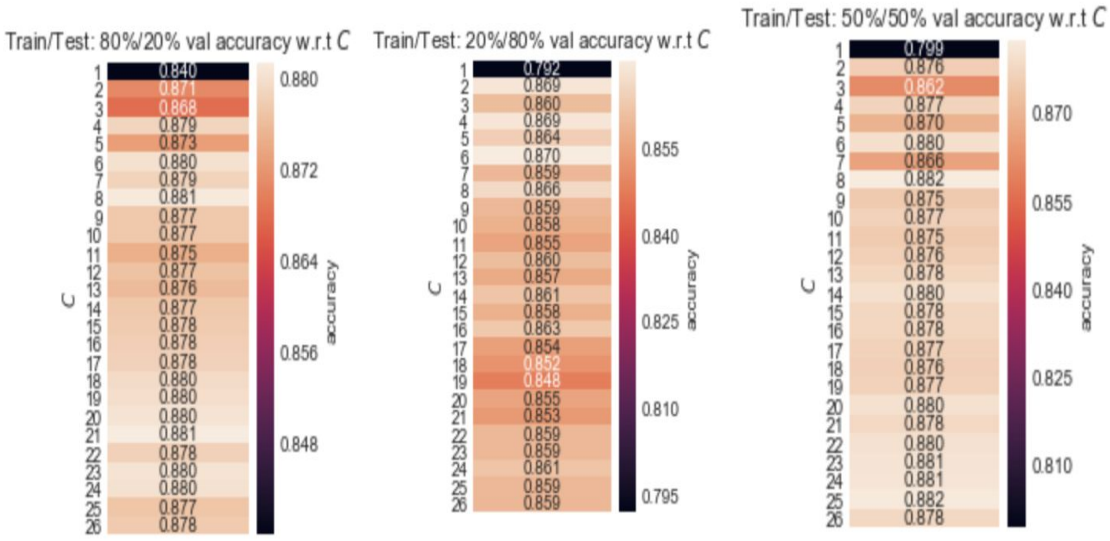
In SVM part, since I changed three parameters: degree, gamma, C in SVM classifier, it is hard to show validation accuracy based on the plot or table. Then I just gave the model and accuracies with optimal parameter.

For 80% training and 20% testing, train accuracy is: 0.90, test accuracy is: 0.88 and the best parameter is degree = 3, radial width =2.00, regularization parameter=1.

For 20% training and 80% testing, train accuracy is: 0.91, test accuracy is: 0.89 and the best parameter is degree = 3, radial width =2.00, regularization parameter=1.

For 50% training and 50% testing, train accuracy is: 0.91, test accuracy is: 0.89 and the best parameter is degree = 3, radial width =2.00, regularization parameter=1.

iii. KNN



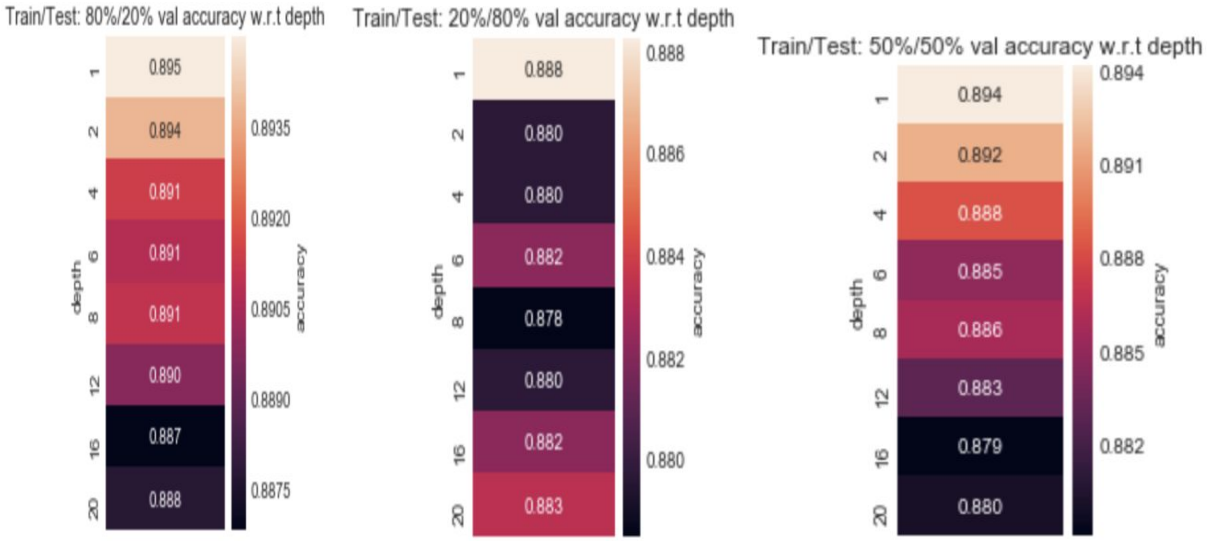
he above of validation accuracies with varying k from 1 to 26 for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train acc is: 0.89, test acc is: 0.89 with C = 21.00.

For 20% training and 80% testing, train acc is: 0.89, test acc is: 0.89 with C = 6.00.

For 50% training and 50% testing, train acc is: 0.89, test acc is: 0.89 with C = 8.00.

iv. Decision Tree



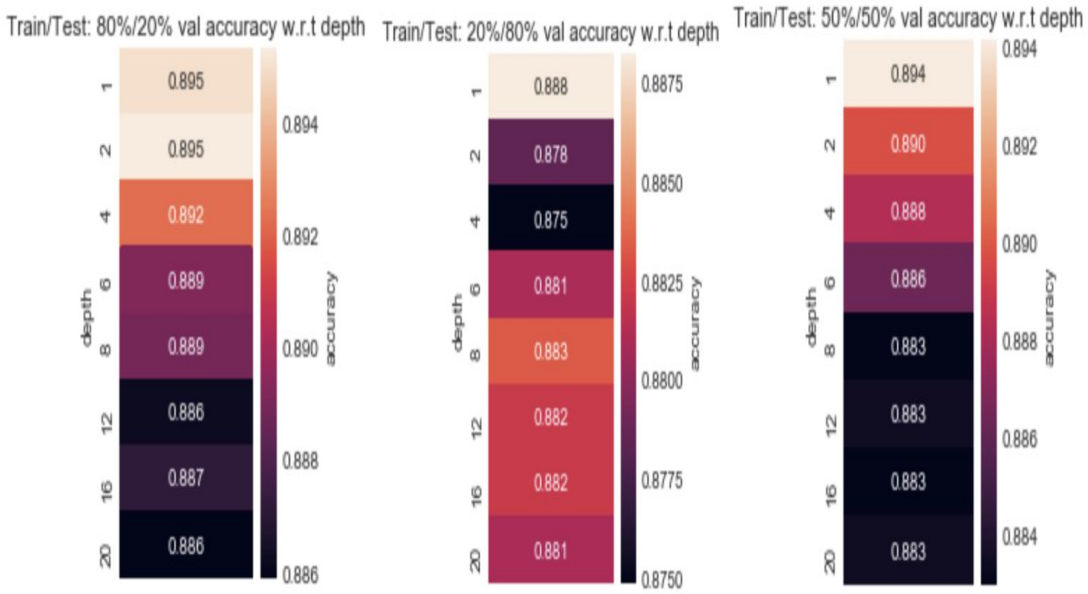
The above of validation accuracies with varying max_depth 1,2,4,6,8,12,16 or 20 for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train accuracy is 0.89 and test accuracy is: 0.88 with max_depth = 1.

For 20% training and 80% testing, train accuracy is 0.89 and test accuracy is: 0.89 with max_depth = 1.

For 50% training and 50% testing, train accuracy is 0.89 and test accuracy is: 0.89 with max_depth = 1.

v. Random Forest



The above of validation accuracies with varying max_depth 1,2,4,6,8,12,16 or 20 for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

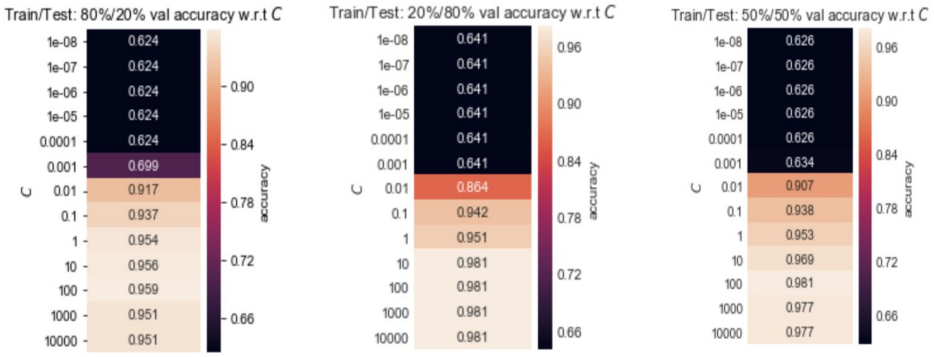
For 80% training and 20% testing, train accuracy is: 1 and test accuracy is: 0.88 with max_depth = 2.

For 20% training and 80% testing, train accuracy is: 0.89 and test accuracy is: 0.89 with max_depth = 1.

For 50% training and 50% testing, train accuracy is: 0.89 and test accuracy is: 0.89 with max_depth = 1.

c. Breast Cancer Dataset

i. Logistic Regression



This is three heatmaps of validation accuracies with varying the ridge (regularation) parameter from 10^{-8} to 10^{4} for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train accuracy is: 0.99, test accuracy is: 0.93 with C = 100.00.

For 20% training and 80% testing, train accuracy is: 0.99, test accuracy is: 0.96 with C = 10.00

For 50% training and 50% testing, train accuracy is: 1.00, test accuracy is: 0.96 with C = 100.00

ii. SVMs

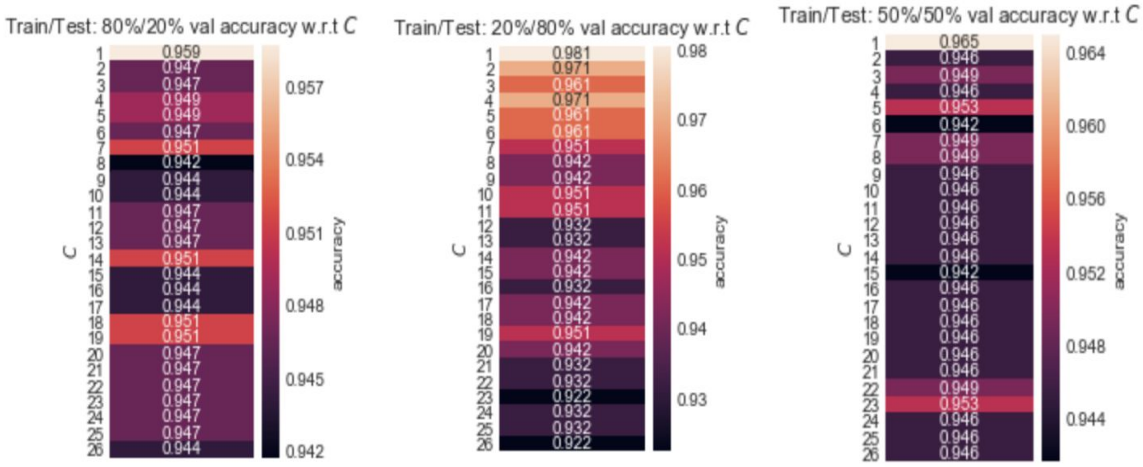
In SVM part, since I changed three parameters: degree, gamma, C in SVM classifier, it is hard to show validation accuracy based on the plot or table. Then I just gave the model and accuracies with optimal parameter.

For 80% training and 20% testing, train accuracy is: 0.90, test accuracy is: 0.89 and the best parameter is degree = 3, radial width =2.00, regularization parameter=1.

For 20% training and 80% testing, train accuracy is: 0.99, test accuracy is: 0.96 and the best parameter is degree = 3, radial width =2.00, regularization parameter=1.

For 50% training and 50% testing, train accuracy is: 0.99, test accuracy is: 0.96 and the best parameter is degree = 3, radial width =2.00, regularization parameter=1.

iii. KNN



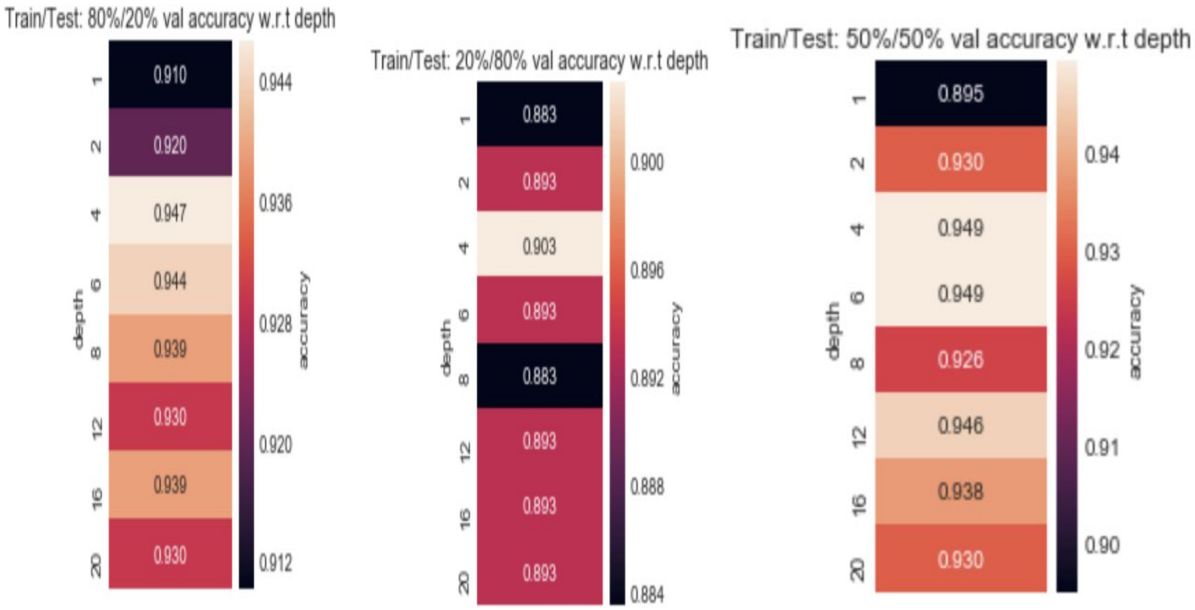
The above of validation accuracies with varying k from 1 to 26 for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train accuracy is: 1.00, test accuracy is: 0.95 with C = 1.00

For 20% training and 80% testing, train accuracy is: 1.00, test accuracy is: 0.95 with C = 1.00

For 50% training and 50% testing, train accuracy is: 1.00, test accuracy is: 0.95 with C = 1.00

iv. Decision Tree



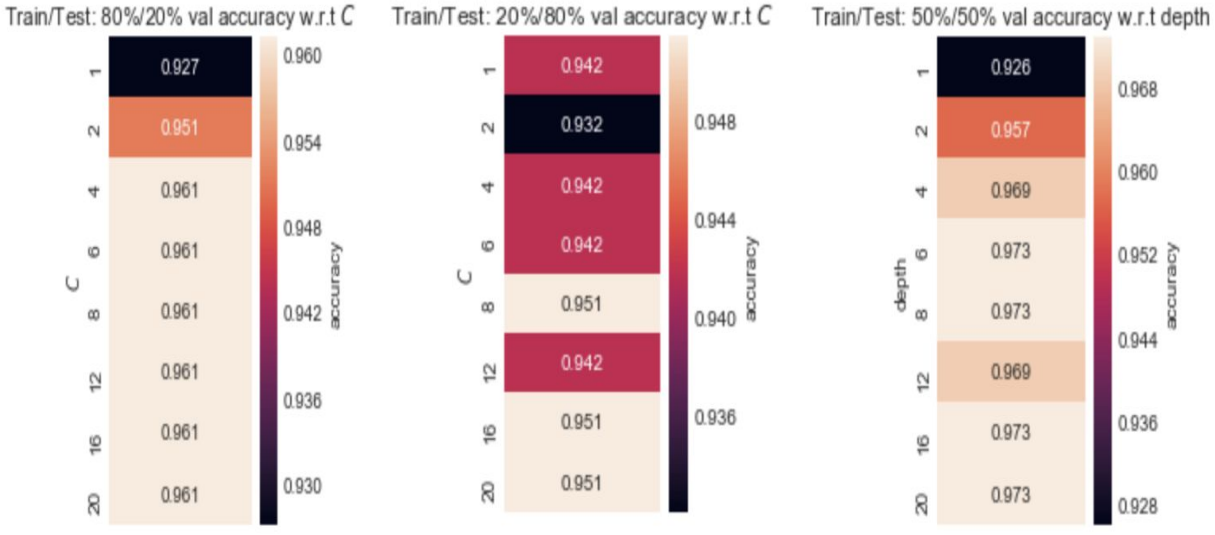
The above of validation accuracies with varying max_depth 1,2,4,6,8,12,16 or 20 for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train accuracy is 0.99 and test accuracy is: 0.92 with max_depth = 8.

For 20% training and 80% testing, train accuracy is 0.99 and test accuracy is: 0.90 with max_depth = 1.

For 50% training and 50% testing, train accuracy is 0.99 and test accuracy is: 0.94 with max_depth = 4.

v. Random Forest



The above of validation accuracies with varying max_depth 1,2,4,6,8,12,16 or 20 for three partitions. We will pick the parameters with the highest validation accuracy. Then we used the hyper-parameter to build up the model. Here is the result:

For 80% training and 20% testing, train accuracy is: 1 and test accuracy is: 0.92 with max_depth = 4

For 20% training and 80% testing, train accuracy is: 1.00 and test accuracy is: 0.94 with max_depth = 8

For 50% training and 50% testing, train accuracy is: 1.00 and test accuracy is: 0.94 with max_depth = 6

6. Conclusion

Test Accuracy for Each Classifier under 3 Partitions for Heart Disease Dataset

Classifier /partition	Logistic Regression	SVM	KNN	Decision Tree	Random Forest
80/20	0.84	0.89	0.84	0.89	0.87
20/80	0.79	0.76	0.79	0.74	0.81
50/50	0.83	0.79	0.80	0.80	0.84

Test Accuracy for Each Classifier under 3 Partitions for Bank Marketing Dataset

Classifier /partition	Logistic Regression	SVM	KNN	Decision Tree	Random Forest
80/20	0.89	0.89	0.89	0.89	0.88
20/80	0.89	0.89	0.89	0.89	0.89
50/50	0.89	0.89	0.89	0.89	0.89

Test Accuracy for Each Classifier under 3 Partitions for Breast Cancer Dataset

Classifier /partition	Logistic Regression	SVM	KNN	Decision Tree	Random Forest
80/20	0.93	0.97	0.95	0.92	0.92
20/80	0.96	0.96	0.95	0.90	0.94
50/50	0.96	0.96	0.95	0.94	0.94

From above three tables, I have three observations. Firstly, normally we would get highest test accuracy when the partition is 80/20 since we used most data to train model and less to test. Secondly, when the sample size is large enough, for example around 4k samples in bank marketing dataset, the test accuracy for each classifier is very close. Third, the random forest and SVM normally perform better, although SVM took a super long time when the data sample is large (more than 1000). One possible reason might be during grid search, I compared too many parameters.

7. References

Caruana, Rich & Niculescu-Mizil, Alexandru. An Empirical Comparison of Supervised Learning Algorithms.

Mangasarian, Olvi & W. Aha, David. UCI repository of machine learning databases.

Pedregosa et al.(2011). Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830.

S. Moro, P. Cortez and P. Rita(2014). UCI repository of machine learning databases.

W. Aha, David. UCI repository of machine learning databases.