Comparison of Support Vector Machine (RBF), K Nearest Neighbor, and Logistic Regression as Supervised Learning Algorithms

Cameron Goharbin
Cognitive Science Undergraduate Student

Abstract

With a plethora of classifier algorithms at one's disposal, each coming with a varying amount of hyperparameters, it can be difficult to decide which algorithm is best suited for a task. This paper will aim to compare and contrast three supervised learning algorithms on three varying datasets with the goal of building an intuition behind the process of selecting the most optimal classifier. The three supervised learning algorithms to be tested in this paper are: support vector machines with radial basis function kernel, k nearest neighbors, and logistic regression.

1. Introduction

The main goal of this paper will be to build an intuition behind how algorithms can be compared. I understand that if I truly wanted to test the performance of different algorithms, a ton more hyperparameters, algorithms, datasets, trials, performance measures, dataset preprocessing methods, etc. This would obviously need a lot of time and computing power that is not accessible to me. SVM with radial basis kernel, KNN, and logistic regression will be the subjects of this test. The performance of these supervised classification algorithms will be measured by their accuracy. Previewing the results, SVM (RBF) performs the best over all data sets, followed by KNN, then logistic regression. This result may not hold too well, however, as their resulting performances are not strictly statistically significant from each other following a p value of less than .05.

2. Methods

2.1. Algorithms

All selected parameter combinations are tested for each algorithm through a five fold cross validation grid search. As mentioned previously, there are many more parameters to be tested if computational efficiency was not a problem.

SVM (RBF) : For SVM I use the radial basis function kernel. The radial width (kernel coefficient) values used are: [.001, .005, .01, .05, .1, .5, 1 2]. For each radial width, I also test the regularization parameter (C) from 10^-8 to 10^4 by factors of 10

KNN: I test 26 values for K neighbors ranging from 1 to 500. Points are uniformly weighted and not weighted different based on distance. The most optimal algorithm to be used to calculate nearest neighbors for each set of data are chosen at run time, and not strictly specified.

Logistic Regression: The regularization parameter (C) for logistic regression will be tested from 10^-8 to 10^4 by factors of ten. The L2 norm is used for penalization.

2.2. Performance Metric

Accuracy is used as the performance metric for this study. Many other performance metrics can be used, however due to the nature of the datasets, post preprocessing, accuracy is a solid metric. Oftentimes it is best to avoid accuracy in the case of having an unbalanced representation of classes, however all datasets used in this paper have fairly equally represented classes.

2.3. Data Sets

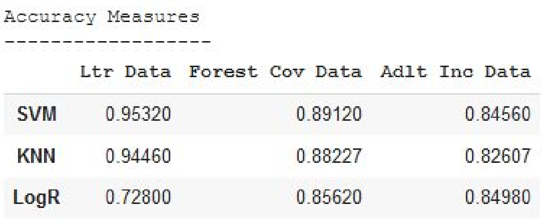
The three datasets used in this paper will be referred to in figures as: Ltr Data, Forest Cov Data, Adlt Inc Data. Two separate tests will be performed on each data set: one using StandardScalar (assuming data stems from normal distribution) to scale the data, one using MinMax (fitting data in a [0,1] range) to scale the data.

Letter data is a data set composed of attributes regarding black and white pixels that make up one of the twenty six letters in the english alphabet. To turn this dataset into a binary classification letters A-M will be considered as the negative class, and letters N-Z will be considered the positive class

Forest cover data is composed of various attributes that can be used to classify different forest cover types. A few of these for example are elevation, soil type, and slope. There are seven cover types in the dataset. To convert this into a binary classification, six of them are turned into the positive class, while the seventh and most abundant in the data set is converted into the negative class.

Adult income data is composed of attributes that might contribute to one's annual income such as, marital status, highest level education, work class, etc. The data set is split into two classes: earning greater than fifty thousand a year and earning less than fifty thousand a year.

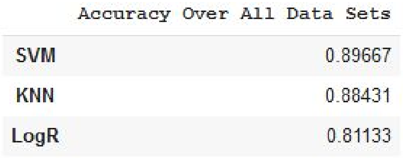
3. Experiment Results (MinMax Scaler)



```
Accuracy Measures
-------------------
```

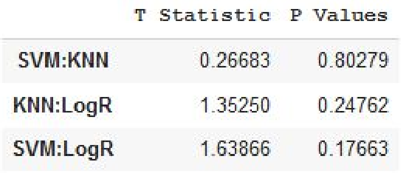
	Ltr Data	Forest Cov Data	Adlt Inc Data
SVM	0.95320	0.89120	0.84560
KNN	0.94460	0.88227	0.82607
LogR	0.72800	0.85620	0.84980

(Table 1. Accuracy score of the best hyperparameter tuned algorithm on each data set)



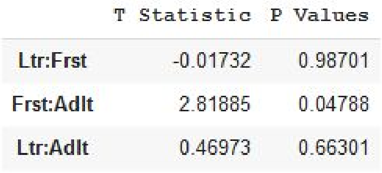
	Accuracy Over All Data Sets
SVM	0.89667
KNN	0.88431
LogR	0.81133

(Table 2. Accuracy score of each algorithm averaged over all data sets)



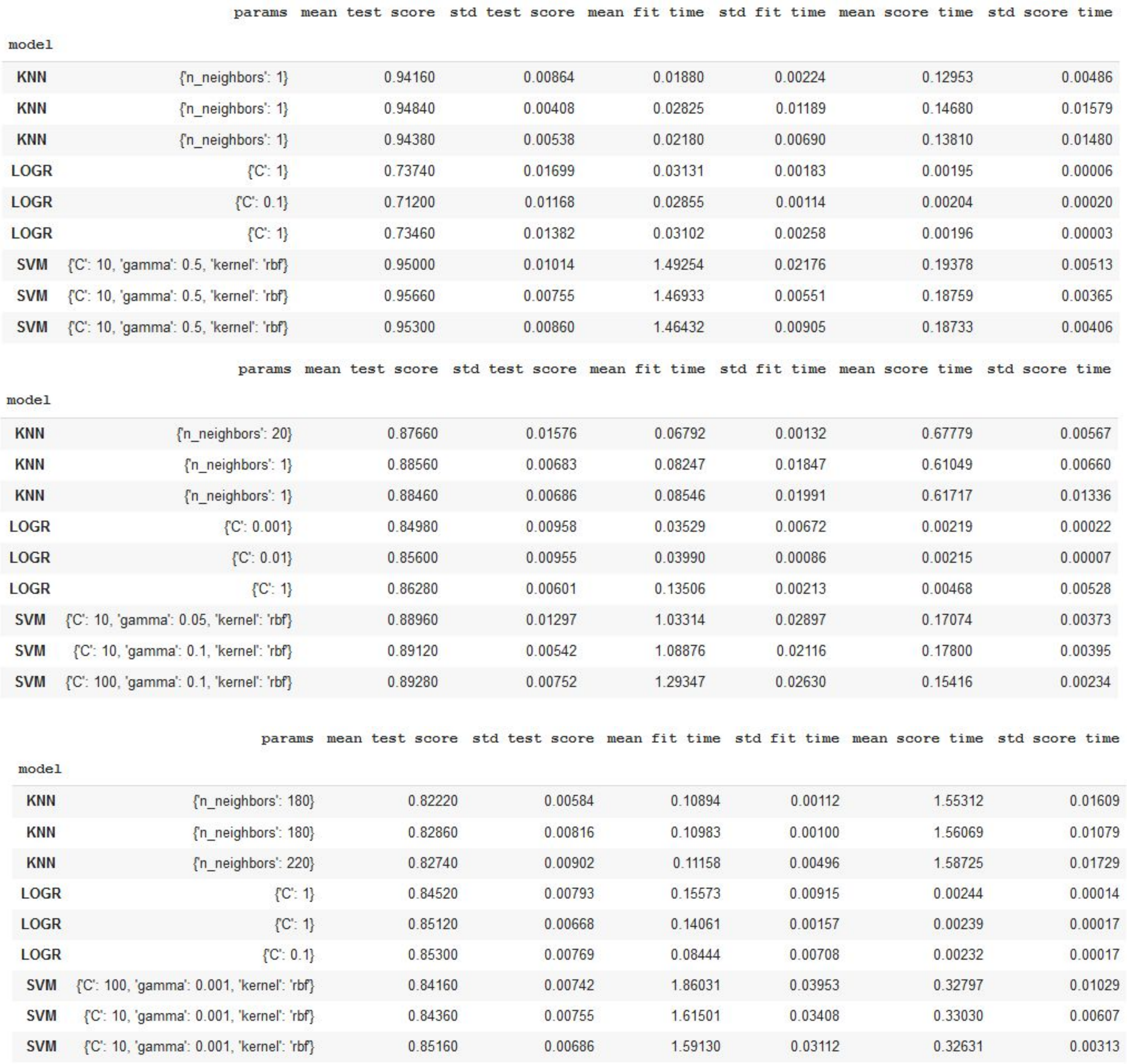
	T Statistic	P Values
SVM:KNN	0.26683	0.80279
KNN:LogR	1.35250	0.24762
SVM:LogR	1.63866	0.17663

(Table 3. T statistic and p value of each algorithm when compared to another)



	T Statistic	P Values
Ltr:Frst	-0.01732	0.98701
Frst:Adlt	2.81885	0.04788
Ltr:Adlt	0.46973	0.66301

(Table 4. T statistic and p value of each data set compared to another)



model	params	mean test score	std test score	mean fit time	std fit time	mean score time	std score time
KNN	{'n_neighbors': 1}	0.94160	0.00864	0.01880	0.00224	0.12953	0.00486
KNN	{'n_neighbors': 1}	0.94840	0.00408	0.02825	0.01189	0.14680	0.01579
KNN	{'n_neighbors': 1}	0.94380	0.00538	0.02180	0.00690	0.13810	0.01480
LOGR	{'C': 1}	0.73740	0.01699	0.03131	0.00183	0.00195	0.00006
LOGR	{'C': 0.1}	0.71200	0.01168	0.02855	0.00114	0.00204	0.00020
LOGR	{'C': 1}	0.73460	0.01382	0.03102	0.00258	0.00196	0.00003
SVM	{'C': 10, 'gamma': 0.5, 'kernel': 'rbf'}	0.95000	0.01014	1.49254	0.02176	0.19378	0.00513
SVM	{'C': 10, 'gamma': 0.5, 'kernel': 'rbf'}	0.95660	0.00755	1.46933	0.00551	0.18759	0.00365
SVM	{'C': 10, 'gamma': 0.5, 'kernel': 'rbf'}	0.95300	0.00860	1.46432	0.00905	0.18733	0.00406

model	params	mean test score	std test score	mean fit time	std fit time	mean score time	std score time
KNN	{'n_neighbors': 20}	0.87660	0.01576	0.06792	0.00132	0.67779	0.00567
KNN	{'n_neighbors': 1}	0.88560	0.00683	0.08247	0.01847	0.61049	0.00660
KNN	{'n_neighbors': 1}	0.88460	0.00686	0.08546	0.01991	0.61717	0.01336
LOGR	{'C': 0.001}	0.84980	0.00958	0.03529	0.00672	0.00219	0.00022
LOGR	{'C': 0.01}	0.85600	0.00955	0.03990	0.00086	0.00215	0.00007
LOGR	{'C': 1}	0.86280	0.00601	0.13506	0.00213	0.00468	0.00528
SVM	{'C': 10, 'gamma': 0.05, 'kernel': 'rbf'}	0.88960	0.01297	1.03314	0.02897	0.17074	0.00373
SVM	{'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}	0.89120	0.00542	1.08876	0.02116	0.17800	0.00395
SVM	{'C': 100, 'gamma': 0.1, 'kernel': 'rbf'}	0.89280	0.00752	1.29347	0.02630	0.15416	0.00234

model	params	mean test score	std test score	mean fit time	std fit time	mean score time	std score time
KNN	{'n_neighbors': 180}	0.82220	0.00584	0.10894	0.00112	1.55312	0.01609
KNN	{'n_neighbors': 180}	0.82860	0.00816	0.10983	0.00100	1.56069	0.01079
KNN	{'n_neighbors': 220}	0.82740	0.00902	0.11158	0.00496	1.58725	0.01729
LOGR	{'C': 1}	0.84520	0.00793	0.15573	0.00915	0.00244	0.00014
LOGR	{'C': 1}	0.85120	0.00668	0.14061	0.00157	0.00239	0.00017
LOGR	{'C': 0.1}	0.85300	0.00769	0.08444	0.00708	0.00232	0.00017
SVM	{'C': 100, 'gamma': 0.001, 'kernel': 'rbf'}	0.84160	0.00742	1.86031	0.03953	0.32797	0.01029
SVM	{'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}	0.84360	0.00755	1.61501	0.03408	0.33030	0.00607
SVM	{'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}	0.85160	0.00686	1.59130	0.03112	0.32631	0.00313

(Table 5. Model performances and hyperparameters from a gridsearchCV for the top performing trials of each algorithm on each dataset (from top to bottom: Ltr Data, Forest Cov Data, Adlt Inc Data)

4. Analysis

As seen in table 1, SVM performs the best for the letter data set and the forest cover dataset, while logistic regression performs the best on the adult income data set. These results do not represent an appropriate ordering of algorithm performance however, due to the obtained t statistics.

With p values of .80279, .24762, .17663 (Table 3), we are unable to declare that any of the algorithms perform significantly different from each other as a function of accuracy. However, we can declare that there was a significant difference in algorithm performance between the forest and adult income data sets (p=.04788). Also, there was no statistical difference in changing the preprocessing method from StandardScaler to MinMax, therefore the results were not even included in the tables from MinMax. In order to find meaningful differences, many more samples would need to be taken. This however would take a large sum of computation power and time and is inaccessible to me for the sake of this paper.

Another large factor to measure an algorithm's performance is run time. Logistic regression was consistently the fastest algorithm (using the previously mentioned hyperparameters) when performing a grid search cross validation to find the hyperparameters.

5. Conclusion

There is a lot that goes into choosing a supervised learning algorithm. It is possible to narrow down the possible options by obtaining an intuition on what algorithms with what hyperparameters are best for different situations. While this paper may have not proven a statistically significant result, I hope it helps build an intuition behind the various thought processes that go behind preprocessing data and fitting an algorithm to a dataset.

6. Conclusion

Repeating my code twice. Once preprocessing with MinMax and the second time with StandardScalar to see if it would make a difference (it didn't).

References

Caruana, Rich & Niculescu-Mizil, Alexandru. (2006). An Empirical Comparison of Supervised Learning Algorithms. Proceedings of the 23rd international conference on Machine learning - ICML '06. 2006. 161-168. 10.1145/1143844.1143865.