COGS 118 – Fall 2018 Final Project
Comparison of Binary Classifying Algorithms

Alberto Montilla
amontill@ucsd.edu

Abstract

"*An Empirical Comparison of Supervised Learning Algorithms*" by Rich Caruana and Alexandru Niculescu-Mizil executes a comparison between ten different supervised learning methods *(1)*. The goal of their study was to analyze the performance difference measured over different parameters of the distinct algorithms. This paper analyzes three of such algorithms, decision trees, KNN, and random forest binary classifiers in an attempt to recreated Caruana and Niculescu-Mizil's results where the classifier's testing accuracy will be used to evaluate performance.

1 Introduction

Binary classifiers are widely used in practical scenarios to predict one out of two labels based on a set of features. For example, one of the studied datasets is used to predict whether or not an office room will be occupied based on light, temperature, humidity, and CO2 level. Classifiers explore data in distinct ways to guess the label of one instance. Such implementations can vary drastically from algorithm to algorithm, resulting in varying performance. The question Caruana and Niculescu-Mizil studied was based around how different classifiers would compare with each other. Being able to replicate their results would not only validate their findings but provide further insight into their method and specific classifier behavior around their parameters, which outlines the objective of this study. Decision trees, KNN, and random forests are the classifiers that will be explored. By changing the dataset structure, as well as parameters for each classifier we will explore when these algorithms work best and how they compare to each other. While classifier performance can be measured from different metrics, this experimentation will focus on the algorithm's testing accuracy.

2 Method

2.1 Data Sets

All datasets used were obtained from the University of California Irvine's Machine learning Repository *(2)*. Three datasets are used in this study, varying in number of instances and number of features (IONOSPHERE *(1)*, TRANSFUSION *(3)*, and OCCUPANCY*(4)*). To prevent biasing produced by features with a large range in the KNN algorithm, each dataset will be normalized before KNN classifier it. For every dataset, we look at our classifier performances when there are whole features missing, meaning that we minimize the feature dimensions to 'force' a more difficult classification which can potentially expose further difference between the classifiers.
IONOSPHERE's labels were transformed to binary values from the original 'g' (good), and 'b' (bad), which describe the quality of radar returns on ionosphere measures. The data has 32 features representing the different values of radar measurements. TRANSFUSION is initially set up for binary classification to determine if an individual donated blood based on past donating behavior. OCCUPANCY is combined and stripped from describing features such as the date each observation was taken.

2.2 Data Partitioning

Every classifier will process each data three times for three different partitions between training and testing values. Such partitions will be 80/20, 50/50, and 20/80, where the first value determines the percentage of instances from the dataset that will be used to train the classifier, and the second value the percentage of instances that will be used during testing. The objective of different partitions is not only to ensure consistency, but to analyze if classifiers will behave differently for each instance.

2.3 Learning Algorithms

The inner implementation of each of the following algorithms is taken from the *sklearn* library *(5)*. Each algorithm has a hyper parameter that is fine-tuned to maximize validation accuracy when testing. The best parameter for each instance will be utilized to test the classifier and obtain a final accuracy. *Max depth* is the parameter to tune for **Decision Tree** and **Random Forest**, and number of neighbors for the **KNN** algorithm.

2.4 Hyper Parameters

An extensive list of parameters is tested per classifier to find the best value for each partition and dataset. While the datasets vary in length and features. The experimenting phase, which will be addressed in detail further below, exposed that after a certain hyper parameter value, the accuracy the classifiers would only decrease, resulting in the list [1, 2, 3, 4, 5, 6, 7, 8, 9, 17, 30, 50] being a comprehensive set of parameters to test where the one that performs the best will be chosen for testing. Essentially prior testing of the parameters and resulting heat maps show that the hyper parameter list above will maximize testing accuracy after the training trials.

2.5 Training Trials

To ensure consistency, for every classifier training 3 trials are executed where the datasets are shuffled and divided into the wanted partition. This ensures that the results found are not outliers, since if only one trial is to be looked at, the results could vary undesirably from training to training. During every trial the classifier is trained on the training partition of the data, where 3-fold grid search cross validation finds the training and validation accuracy for every parameter in the hyper parameter list of values to be tested per classifier. From there, the best parameter is found and chosen to be used on the testing stage. When testing, the classifier fits the test data based on the partition and best parameter, and calculates the final accuracy

3 Experiment

The experimenting phase is comprised by each classification algorithm processing each dataset for every (80/20, 50/50, 20/80) partition, where in each training the best hyper parameter will be chosen to test the resulting classifier. Below are the results of the experiment for each dataset, in ascending dataset size order.

3.1.1 IONOSPHERE

This dataset is comprised of 351 instances and 32 features.

	Decision Tree		*KNN*		*Random Forest*	
Partition	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy
80/20	.894	.944	.871	.986	.919	1.0
50/50	.889	.960	.891	.994	.927	.988
20/80	.876	.900	.876	.989	.890	.996

Experimentation on reduced features, 351 instances and 1 feature.

	Decision Tree		*KNN*		*Random Forest*	
Partition	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy
80/20	.845	.873	.849	.901	.844	.901
50/50	.823	.841	.796	.875	.813	.875
20/80	.848	.851	.795	.808	.762	.847

Findings: The test accuracy of the classifiers is best with random forest classifier and worst with decision tree classifier. While the accuracies are very high for both experiments, when the

features are reduced a considerable drop in accuracy is seen, However, the accuracy for KNN is now closer to that of random forest.

3.1.2 TRANSFUSION

This dataset is comprised of 748 instances and 4 features.

	Decision Tree		*KNN*		*Random Forest*	
Partition	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy
80/20	.895	.958	.874	.986	.930	1.0
50/50	.867	.943	.836	.989	.918	.983
20/80	.838	.996	.838	.978	.933	.989

Experimentation on reduced features, 748 instances and 2 features.

	Decision Tree		*KNN*		*Random Forest*	
Partition	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy
80/20	.769	.807	.770	.847	.766	.813
50/50	.764	.781	.768	.807	.761	.741
20/80	.729	.746	.752	.789	.745	.756

Findings: The test accuracy of the classifiers is best with random forest classifier and worst with decision tree classifier. While the accuracies are very high for both experiments, when the features are reduced a considerable drop in accuracy is seen. In this case KNN accuracy when the features are reduced actually surpassed that of random forest.

3.1.2 OCCUPANCY

This dataset is comprised of 20560 instances and 5 features

	Decision Tree		*KNN*		*Random Forest*	
Partition	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy
80/20	.991	.998	.993	1.0	.991	.999
50/50	.990	.989	.992	.993	.991	.996
20/80	.989	.998	.989	.991	.989	.995

Experimentation on a fourth of the data, 5140 instances and 5 features.

	Decision Tree		*KNN*		*Random Forest*	
Partition	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy
80/20	.990	.994	.989	.990	.991	.999
50/50	.990	.989	.989	1.0	.989	.993
20/80	.992	.990	.990	.992	.990	.997

Experimentation on reduced features, 20560 instances and 1 features.

	Decision Tree		*KNN*		*Random Forest*	
Partition	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy
80/20	.788	.814	.815	.991	.796	.820
50/50	.788	.797	.799	.887	.789	.806
20/80	.781	.781	.784	.833	.783	.799

Experimentation on reduced features and a fourth of the data, 5140 instances and 1 feature.

	Decision Tree		*KNN*		*Random Forest*	
Partition	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy	Validation Accuracy	Test Accuracy
80/20	.792	.792	.786	.862	.798	.849
50/50	.797	.789	.783	.835	.793	.803
20/80	.789	.792	.790	.829	.786	.787

Findings: This dataset is much larger than the other two, which yields a high accuracy for all three classifiers. When features are reduced, accuracy not only drops, but that of KNN surpasses that of random forest even though the opposite occurred previously. In both cases, when the number of instances is reduced by a factor of four, the test accuracies drop, more significantly in the last case where both feature dimension and dataset size are reduced.

3.2 Combined findings

Combining the above results, we attain the following means:

No feature reduction

Classifier	*IONO*	*TRANS*	*OCCUP*	*MEAN*
RF	.995	.991	.997	**.994**
KNN	.990	.984	.994	.989
DT	.935	.966	.995	.965

Feature reduction

Classifier	*IONO*	*TRANS*	*OCCUP*	*MEAN*
RF	.874	.770	.808	.817
KNN	.861	.814	.904	**.856**
DT	.855	.778	.797	.810

Partition analysis

Partition	*Validation Accuracy (mean)*	*Testing Accuracy (mean)*
80/20	**.929**	**.986**
50/50	.922	.981
20/80	.913	.971

Data size analysis

OCCUPANCY Dataset	*Validation Accuracy (mean)*	*Testing Accuracy (mean)*
Large(20560)	**.891**	**.916**
Small(5140)	.890	.904

As seen above, validation accuracy and therefore testing accuracy are highest in the 80/20 partition ad lowest in the 20/80 partition, which confirms the expectation that the larger the training and validation set are, the greater validation accuracy will be since the classifiers will be trained on more data.
OCCUPANCY's data size variation exposes that with less instances, our accuracy will decrease. When reducing the number of data set features, the performance of KNN matched and even became greater than that of random forest. This exposes that random forest can be more affected by feature reduction than KNN.

4 Conclusion

Comparing our findings with those of Caruana and Niculescu-Mizil, a clear similarity is observed, where random forest classifier performs the best in front of KNN classifier and lastly decision tree classifier. Our experiment showed that the larger the training set is in our partition, the better validation and test accuracy we will obtain. Additionally, we explored how feature reduction may vary our results, where accuracy is not only reduced, but more significantly in random forest classifier than in KNN. In general, we were able to replicate the conclusions in Caruana and Niculescu-Mizil and extended our analysis from such conclusions. While random forest is the better performing classifier, KNN is very close in accuracy and if feature loss becomes a problem, KNN classifier could result in being more accurate than random forest. In either case, decision tree classifier performed worse than the other two.

References

(1) Caruana, Rich, and Alexandru Niculescu-Mizil. “An Empirical Comparison of Supervised Learning Algorithms.” Proceedings of the 23rd International Conference on Machine Learning - ICML 06, 2006.

(2) Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

(3) Yeh, I-Cheng, Yang, King-Jang, and Ting, Tao-Ming, "Knowledge discovery on RFM model using Bernoulli sequence, "Expert Systems with Applications, 2008,

(4) Luis M. Candanedo, Vãeronique Feldheim. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Energy and Buildings. Volume 112, 15 January 2016, Pages 28-39.

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