Abstract:

There are many types of machine learning algorithms and all of them have their strengths and weaknesses. This paper compares three of them: Decision Trees, Random Forest Classifiers, and Support Vector Machines (SVMs) using datasets from the UCI Machine Learning website. This paper also studies different partitioning techniques in order to determine which is the best split for training and testing data. It was found that the best split for training and testing data was to have a greater amount of training data than testing data. It was also found that Decision Trees and Random Forest Classifiers performed better than SVMs on the datasets studied.

Introduction:

This project is an empirical study on different types of machine learning algorithms that were covered in the Cogs 118A Machine Learning class. Often times it is hard to compare machine learning algorithms learned in lecture because the homeworks cannot be entirely comprehensive. This project provided the author with the chance to pick and choose their own algorithms and datasets to fully engage with the models presented in class. This project was looking at how different machine learning models perform on different sized datasets. The goal was to either support or contradict some of the results in Caruana and Niculescu-Mizil’s paper: “An Empirical Comparison of Supervised Learning Algorithms”. In particular, it was expected that when partitioning the data between testing and training, algorithms would be more accurate when the amount of training data was greater than or equal to the amount of testing data. It was also expected that Random Forests would be more accurate than SVMs which would be more accurate than Decision Trees given that those were the results of Caruana and Niculescu-Mizil’s paper.

Methods

Learning Algorithms Used:

Machine learning algorithms were picked that would perform differently on the datasets to show their different strengths. Additionally, these algorithms were picked because they are fast and do not take a huge amount of computational time as the analysis was run on a laptop.

SVM: SVM with a linear kernel was included because it is a high performer compared to other machine learning algorithms.

Decision Tree: Decision Tree was included because it is usually one of the worse machine learning algorithms and it was interesting to test to see if this were the case with all the datasets.

Random Forest Classifier: Random Forest(RF) Classifier was included to test to see if they perform better than Decision Tree algorithms since RFs are an aggregation of decision trees.

Comparing Across Performance Metrics:
The performance metric used was accuracy on the test dataset.

Calibration Metrics:
For each machine learning algorithm used, there were hyperparameters that were tested.
For the SVM model, the hyperparameter was the C value, the penalty parameter of the error term. According to the SciKit-Learn website, the noisier the data, the smaller this parameter should be. The list of C values tried was [0.00001, 0.0001, 0.001, 0.01, 1].
For both the Decision Tree model and the Random Forest model, the hyperparameter used was the max-depth value. The range tested was [1, 2, 3, 4, 5, 6]. These models are both prone to overfitting, so the maximum depth needs to be kept short.

Cleaning and Prepping the Data Sets:
1. The first dataset used was the Mushroom Dataset from the UCI Machine Learning website. There are 22 attributes in this dataset which are all used to classify a poisonous mushroom or an edible mushroom. The features were classifications like 'cap-size' and 'gills'. Several rows were dropped in this dataset as there was missing data. The number of datapoints for one classification was around 3500 and for the other classification there were around 2160 data points. One-hot encoding was used to prep the data for the machine learning algorithms as all the features were categorical data.
2. The second dataset was the Breast Cancer Wisconsin Dataset. This dataset predicts malignant tumors versus benign tumors. There are 8 attributes in this data. This dataset also had a few rows that were dropped because of missing data. The number of datapoints for one classification was around 400 data points for one classification and 200 data points for the other classification. This is a much smaller dataset than the first so it was suspected that the machine learning models would all perform worse with this dataset.
3. The third dataset was the Contraceptive Dataset. It has three classifications, whether a woman used no birth control, short-term birth control, or long-term birth control. It uses 9 features to predict those labels. Because there were three classifications, the long-term birth control class was dropped. This meant that there were around 630 data points for one classification and 222 data points for the other classification. This dataset was also small compared to the first dataset so it is suspected that the classification would be worse.

4. The fourth dataset used was the Adult Dataset. This dataset was collected census data which classified whether or not someone earned above or below 50k. The dataset was chosen to contrast with the other datasets because it is the largest dataset, with around 30,000 data point. There were also many attributes in this dataset; with one-hot encoding, the number of attributes was around 113.

**All datasets were found on the UCI Machine Learning website*

Implementation of Machine Learning Algorithms:

All the machine learning models were drawn from the Scikit-Learn website. Besides using their methods for Decision Tree, Random Forest, and SVM, the GridSearchCV method was also used. The GridSearch method tested the hyperparameters and also performed the cross validation and 3 k-fold techniques with the classifiers.
For each dataset, the following steps were performed:

1. Choosing a classifier
2. Splitting up the data into testing and training
3. Initializing a classifier
4. Initializing a GridSearch with the list of hyperparameters
5. Training the classifier using the output of the GridSearch
6. Drawing a heatmap of the hyperparameters
7. Training a new classifier with the hyperparameter that returned the highest accuracy
8. Using the new classifier to predict labels on the test set
9. Repeat 1 - 8 for each classifier
10. Repeat 9 for each testing and training partition

Results:

The following tables shows the accuracy (rounded to the third decimal) of the machine learning models on each dataset as well as the average accuracy for each model and testing/training partition.

Accuracy Results of Machine Learning Algorithms on Cancer Dataset

	80% Train 20% Test	**50% Train 50% Test**	**20% Train 80% Test**	**Average Model Accuracy**
SVM	0.985	0.985	0.965	0.978
Decision Tree	0.971	0.962	0.948	0.960
Random Forest	0.978	0.985	0.965	0.976
Average Accuracy for Partition	0.978	0.977	0.959	

Accuracy Results of Machine Learning Algorithms on Mushroom Dataset

	80% Train 20% Test	**50% Train 50% Test**	**20% Train 80% Test**	**Average Accuracy for Model**
SVM	0.926	0.867	0.579	0.777
Decision Tree	0.954	0.956	0.579	0.829
Random Forest	0.902	0.848	0.579	0.776
Average Accuracy for Partition	0.927	0.890	0.579	

Accuracy Results of Machine Learning Algorithms on Contraceptive Dataset

	80% Train 20% Test	**50% Train 50% Test**	**20% Train 80% Test**	**Average Accuracy for Model**
SVM	0.672	0.677	0.688	0.679
Decision Tree	0.688	0.677	0.736	0.700
Random Forest	0.698	0.719	0.730	0.716
Average Accuracy for Partition	0.686	0.691	0.718	

Accuracy Results of Machine Learning Algorithms on Adult Dataset

	80% Train 20% Test	**50% Train 50% Test**	**20% Train 80% Test**	**Average Accuracy for Model**
SVM	*	*	*	*
Decision Tree	0.843	0.839	0.840	0.841
Random Forest	0.820	0.812	0.823	0.818
Average Accuracy for Partition	0.831	0.825	0.831	

*SVM did not finish in a time that was reasonable for this dataset.

Conclusions:

The two methods that were being tested were how partitioning up the data affected accuracy and how different machine learning models performed on the datasets.

Partitioning Discussion:

The results for how partitioning the data affected accuracy were clear; 80% training and 20% testing performed the best, with 50% testing and 50% training following close behind and 20% training and 80% testing performing the worst. 80% training and 50% training were relatively close to each other, while 20% training was far behind. This supports the initial reasoning that it is better to have more training data than test data.

Machine Modeling Discussion:

The results for the machine modeling comparison were less clear. Decision Trees and Random Forests were tied for the most accurate, and SVMs came up last. This was surprising as Decision Trees were expected to perform the worst. It is suspected that this has to do with how different the datasets were from each other and one of the datasets(the Adult dataset) was so large that the SVM algorithm never finished running. The consistent top performer was the Random Forest Classifier, but the Decision Tree performed much better than expected. So although the partitioning results were as expected, the results from the paper by Caruana and Niculescu-Mizil were unable to be exactly reproduced.

Justification for Bonus Points:

The justification for bonus points in this project is that four datasets were studied instead of three and the last dataset was much larger than the rest. This increased the relevancy of the results since a diverse set of data was studied.

References:

Caruana, Rich, and Alexandru Niculescu-Mizil. “An Empirical Comparison of Supervised Learning Algorithms.” *Https://Www.cs.cornell.edu*, Cornell, 2006, www.bing.com/cr?IG=A648D49C8A4C4D4F886E544DFDC94E6A&CID=2DEA4C9BC749624E0FCA4727C6EF6321&rd=1&h=zrBtzR-2ZkoEojJOWFJdUKY7HLRwNoNizzNi37gPzJk&v=1&r=https%3a%2f%2fwww.cs.cornell.edu%2f%7ecaruana%2fctp%2fct.papers%2fcaruana.icml06.pdf&p=DevEx,5066.1.

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