CS 7641 Assignment #1: Supervised Learning

William Koppelman
wkoppelman3@gatech.edu

I. Datasets

A. *Contraceptive Method Choice*

This dataset is a subset of a 1987 contraceptive prevalence survey taken in Indonesia. It is a survey of married women who were not pregnant. The goal of the survey was to predict current contraceptive use (no use, short-term, or long-term methods) based and demographic and socio-economic characteristics of the women and her husband. This is a ternary classification dataset with 9 attributes and 1,473 instances. More information can be found about the dataset in the README.

This is an interesting dataset because it looks at the effect of different areas of influence in the person's life to determine a very personal choice. Religion, education, social class, and the generation in which they were raised are all being considered.

From a data analysis standpoint it is interesting because it could be a binary classification but they chose to make it ternary. It also includes numerical as well as ordinal and categorical data. This will help to test the abilities and strengths of each supervised learning method.

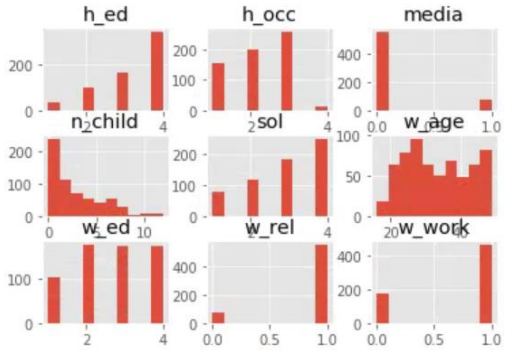

Fig. 1. Histogram of the nine attributes within the no contraceptive class

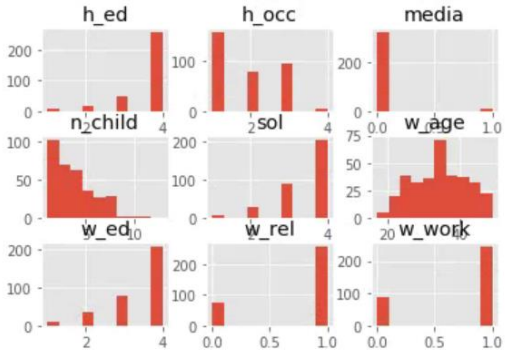

Fig. 2. Histogram of the nine attributes within the long-term use class

The most notable things from the exploratory data analysis (Fig. 1-3) is that a higher education among the women and their husbands appears to lead to contraceptive use. Also, those that use contraception appear to be from a younger generation. However, the converse does not appear to be true as all ages of women are in the no contraceptive use class.

B. *Red Wine Quality*

This dataset is a collection of physicochemical attributes on the ordinal sensory output of quality. It only looked at Portuguese "Vinho Verde" red wine. There are 11 attributes and 1,599 instances in this dataset. The quality score ranges from 0 to 10 but only scores between 3 and 8 are present in the data. More information can be found about the dataset in the README.

I found this dataset to be interesting because it looks at the very subjective reason certain wines are preferred on a chemical level. In a multi-billion dollar business, wine makers would look for any advantage possible to earn more money. Some of these chemicals are also the center of current marketing fads, so it would be interesting to see if there is any research to back them up.

I chose this as interesting from a machine learning point of view because it has an ordinal quality score as its classification. It also included only continuous attributes in contrast with the contraceptive use dataset which had mainly ordinal or categorical attributes.

As you can see from Figure 4, certain attributes vary more with the score than others. The last tile is a histogram of the wine quality scores. As expected, the majority of the scores fall within the middle of the range.

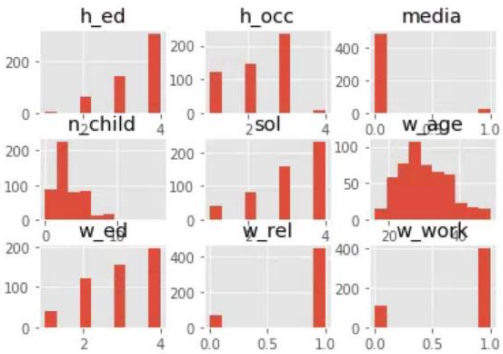

Fig. 3. Histogram of the nine attributes within the short-term use class

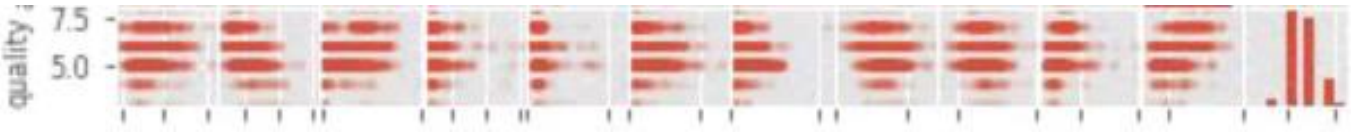

Fig. 4. Scatterplot of the 11 attributes based on wine quality

II. Supervised Learning Algorithms

A. *Decision trees with pruning*

Starting with an unpruned decision tree we can see in the case of contraceptive use and red wine quality that both tend to overfit the data (Table I). This is expected because the tree will keep splitting until it finds the answer that works best. Having too many dimensions can lead to overlap and pointless regularity, thus ignoring the distinctive features.

Another point to note is that the training and testing time was consistently longer with the contraception dataset, even though it contains fewer samples. This can most likely be attributed to the different data types with this set (quantitative and qualitative).

To implement pruning using the *scikit learn* module in python, I performed a grid search over a variety of factors that each effectively prune the tree in some one. These hyperparameters were the minimum samples per split, minimum samples per leaf, maximum number of features to include, maximum depth, and a splitting criterion. The best performing cross-validated, hyperparameters are shown in Table II.

As expected, the pruned decision trees had a decreased training accuracy (Table I). While the testing accuracy improved. The contraception tree was pruned in all of the hyperparameters while the red wine decision tree only pruned the maximum features and depth. Pruning the maximum features shows us that there may be some confounding variables.

TABLE I. Decision Tree Statistics

Model	Training Accuracy	Testing Accuracy	Training Time	Testing Time
Contraception Unpruned	0.962	0.488	0.0581	0.0124
Contraception Pruned	0.556	0.580	0.0105	0.0025
Red Wine Unpruned	1.000	0.575	0.0164	0.0017
Red Wine Pruned	0.557	0.597	0.0150	0.0012

TABLE II. Decision Tree Hyperparameters

Model	Min samples per split	Min samples per leaf	Max features	Max depth	Split Criterion
Contraception	4% (n=59)	7% (n=104)	90% (n=8)	6	Entropy
Red Wine	n=2	n=2	80% (n=8)	13	Gini

When looking at the confusion matrices for these trees we notice a couple of things. In Table III we can see that the model does a much better job classifying the no contraception use (first row). I expect this has to do with the other two rows having some overlap in features. However, in the last two rows (long-term and short-term contraceptive use), they are often misclassified as no contraceptive use. This means the tree is having a hard time distinguishing their characteristics.

In Table IV we can see that the red wine confusion matrix predicts the majority of samples correctly along the diagonal. Upon closer inspection, more than 94% of the samples are either on the diagonal or only off by one spot. The model does seem to have difficulty classifying the lowest score (first column/row) and there also seems to be some similarity and crossover with the middle scores (center of the matrix). This shows the similarity and subjectiveness of taste.

TABLE III. Contraceptive Use Decision Tree Confusion Matrix

90	11	18
25	27	21
34	15	54

TABLE IV. Red Wine Decision Tree Confusion Matrix

0	0	0	1	0	0
1	4	6	2	0	0
3	4	87	41	4	0
0	3	31	79	19	2
0	0	3	7	20	0
0	0	0	0	1	1

Fig. 4 and 5 show the learning curves for the hypertuned decision trees. Fig. 4 for the contraceptive use is what you hope for as the pruning has found the optimal balance between bias and variance.

Fig. 5 shows the general direction of training error is decreasing but the testing error is not improving. It appears that a couple of things could be going on here. The training curves could converge if more training samples were available to add. However, there appears to be a case of low bias/high variance in the model. This is common in decision trees. Even though the grid search of hyperparameters found these as the best, we may still need to better prune the tree.

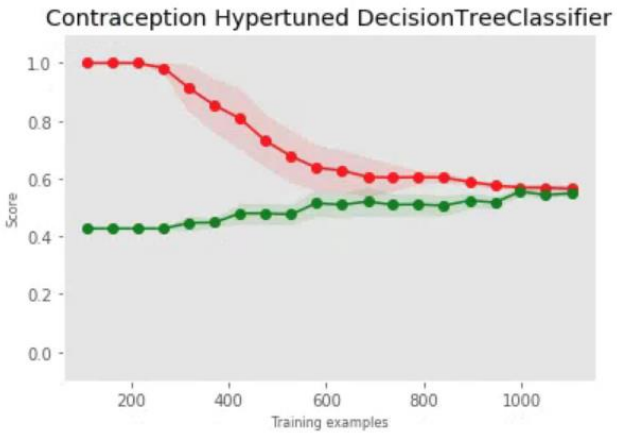

Fig. 5. Contraceptive use decision tree learning curve

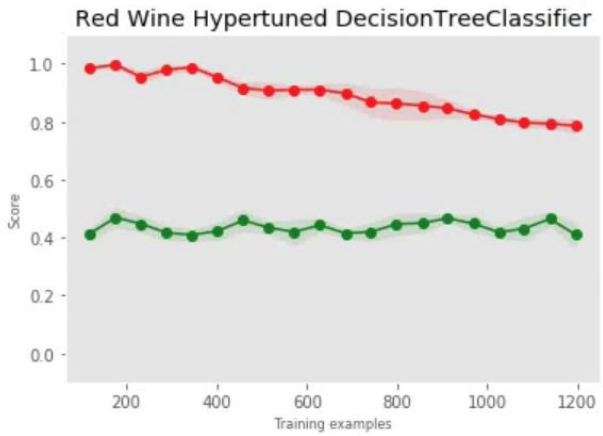

Fig. 6. Red wine decision tree learning curve

B. *Neural networks*

The neural network provided an interesting modeling experience because it seemed that choosing the appropriate structure for the network was very important.

The results are shown in Table V. As expected, the training times take longer. However the testing accuracy is not as high as the decision tree.

When I began building my initial network I looked at the number of layers to add to it. Testing for both datasets led to three layers. Once I had this structure set up for my model I was able to tune it using batch size, epochs, optimizer, activation kernel, nodes, and initialization mode. The optimal parameters can be seen in Table VI. While I feel confident in these optimal parameters, I feel like the structure of the network with the number of layers and neurons in each layer could use some further consideration.

TABLE V. NEURAL NETWORK STATISTICS

Model	Training Accuracy	Testing Accuracy	Training Time	Testing Time
Contraceptive use	0.561	0.536	11.2811	0.1707
Red Wine	0.613	0.553	26.2831	0.7943

TABLE VI. NEURAL NETWORK HYPERPARAMETERS

Model	Activation	Batch size	Epochs
Contraceptive use	Softplus	10	100
Red wine	Softplus	10	150
	Initial mode	**Neurons**	**Optimizer**
Contraceptive use	he_uniform	10	RMSprop
Red wine	uniform	8	Nadam

Looking at the contraceptive use learning curves for our neural network (Fig. 8) shows the model accuracy/loss for the training and testing data start to diverge a little after around epoch 30. This appears to be where overfitting begins to occur.

The red wine learning curve (Fig. 7) shows a high variability in the testing data. I researched online and asked in the class discussions but could not determine a reason for this. The training and testing data seem to diverge around epoch 60, again appearing to overfit the data after this point.

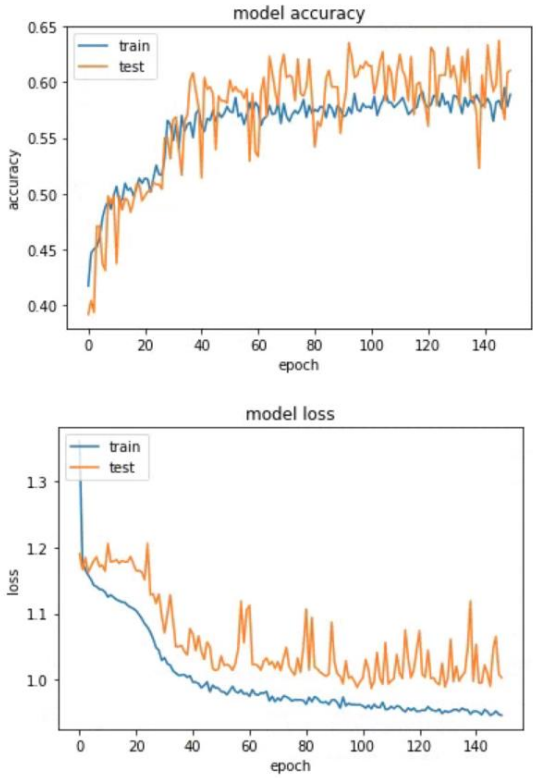

Fig. 7. Red wine neural network learning curve

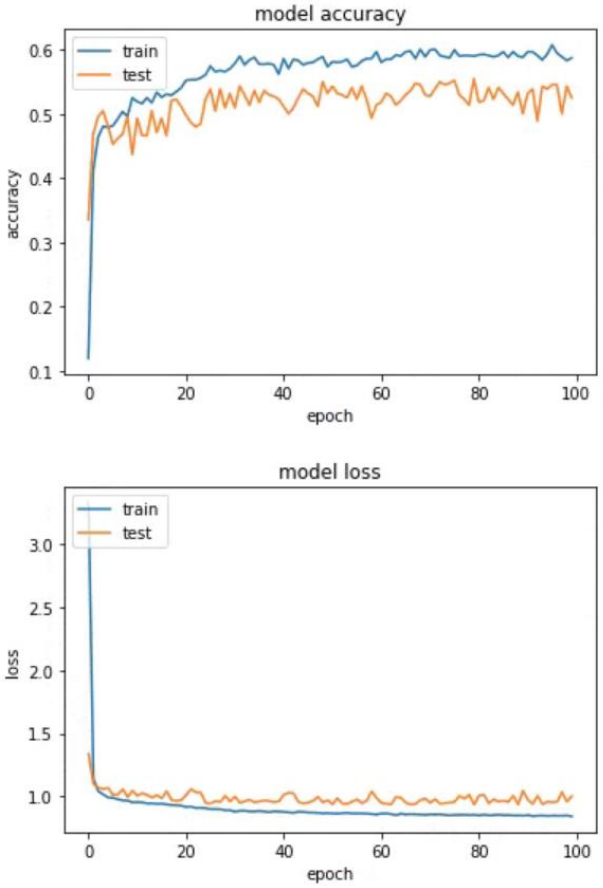

Fig. 8. Contraceptive use neural network learning curves

C. *Boosting*

For boosting I implemented to *XGBoost* classifier on decision trees. *XGBoost* is a gradient boosting library that is fast because it allows for parallel tree boosting. It also connects easily to the *scikit-learn* modules for easy training, testing, and optimizing.

The testing accuracy on the contraceptive use dataset actually decreased marginally (Table VII). However, the red wine dataset saw an improvement of around 6% accuracy. This is what was expected. The lack of improvement in the contraceptive use dataset was surprising. The two factors that I could come up with for this were the overlap in some of the attributes and the overlap in some of the contraceptive use classes.

The tuned hyperparameters (Table VIII) are pretty similar to the original decision tree parameters. With the contraceptive use model having a larger alpha value, it is doing some feature selection. This lends me to believe that the output classes are a little confounding.

TABLE VII. BOOSTING STATISTICS

Model	Training Accuracy	Testing Accuracy	Training Time	Testing Time
Contraceptive use	0.559	0.573	0.0710	0.0035
Red wine	0.693	0.659	1.3539	0.0210

TABLE VIII. BOOSTING HYPERPARAMETERS

Model	Sub-sample	Alpha	Estimators	Max depth
Contraceptive use	95%	1	25	4
Red Wine	85%	1e-7	105	10
Model	**Learning Rate**	**Gamma**	**Column sample by tree**	
Contraceptive use	0.4	0.25	90%	
Red Wine	0.06	0.01	90%	

The confusion matrices show some improvement on the pruned decision trees. The contraceptive use for the boosting model (Table IX) does much better with short-term contraceptive use (third row) but the long term use (second row) appears to be no better than chance.

The red wine boosting confusion matrix (Table X) not only improved the overall accuracy but also included more on the off diagonal (only misclassifying by one point). Around 96% of the testing set is on the diagonal or off diagonal.

TABLE IX. CONTRACEPTIVE USE BOOSTING CONFUSION MATRIX

84	9	26
21	24	28
25	17	61

TABLE X. RED WINE BOOSTING CONFUSION MATRIX

0	0	1	0	0	0
0	0	8	5	0	0
0	2	96	31	2	0
0	1	28	94	8	0
0	0	1	17	21	0
0	0	0	3	2	0

The learning curve for the contraceptive use boosting model (Fig. 9) shows that the model is learning but does not completely close the gap in the training and testing data. This can be caused by high variance and low bias. More data would help to improve this.

The learning curve for the red wine boosting model (Fig. 10) shows a training set that has overfitted the data and not much learning with the testing set. Since the model showed improvement on the original pruned decision tree I did not expect this. The best conclusion I could come to after researching and discussing on the message boards was that was that pruning did not provide improvements in the model accuracy. You can see from the model parameters that it is not heavily pruned.

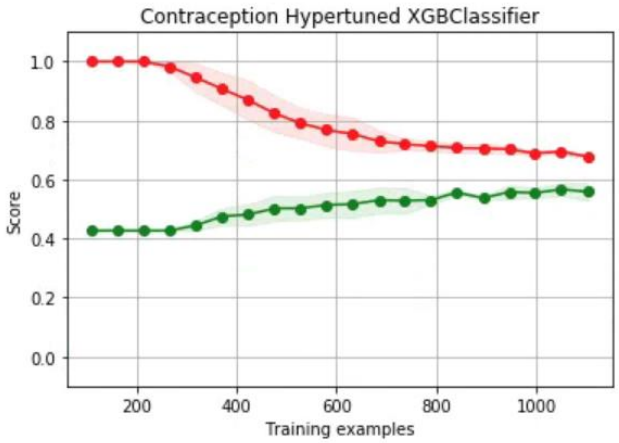

Fig. 9. Contraceptive use boosting learning curve

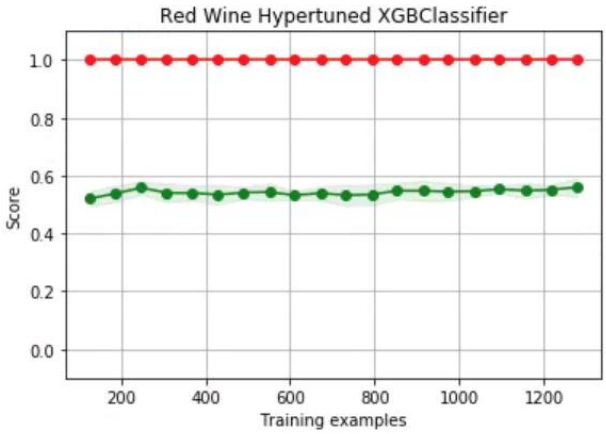

Fig. 10. Red wine boosting learning curve

D. *Support vector machines*

For the support vector machine model I used the *scikit-learn SVC* function. This uses C as a penalty term and allows you to choose a kernel. Its main drawback is that it scales quadratically and is tough to use on a dataset of larger than 10,000 samples. However, both of mine were an order of magnitude under that and I had no issues.

The testing accuracy was similar to the other models and training and testing times were reasonable (Table XI).

For the hyperparameters I used a linear, sigmoid (as discussed in the lectures), and *Radial Basis Function* (RBF) kernel and I varied the penalty parameter C and gamma (for sigmoid and RBF) logarithmically. The grid search found the optimal hyperparameters, which are listed in Table XII.

TABLE XI. SUPPORT VECTOR MACHINE STATISTICS

Model	Training Accuracy	Testing Accuracy	Training Time	Testing Time
Contraceptive use	0.555	0.576	0.1178	0.0097
Red wine	0.651	0.644	0.1272	0.0160

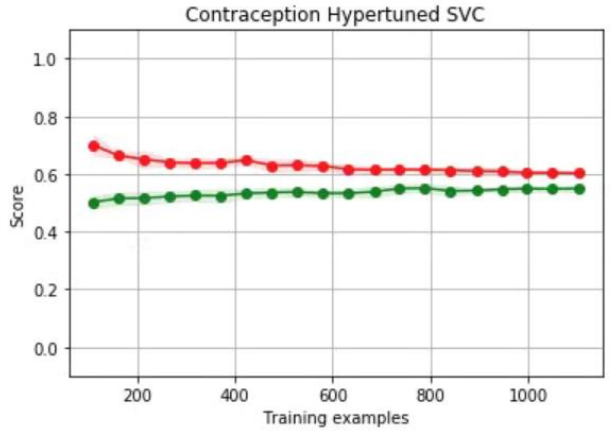

Fig. 11. Contraceptive use support vector machine learning curve

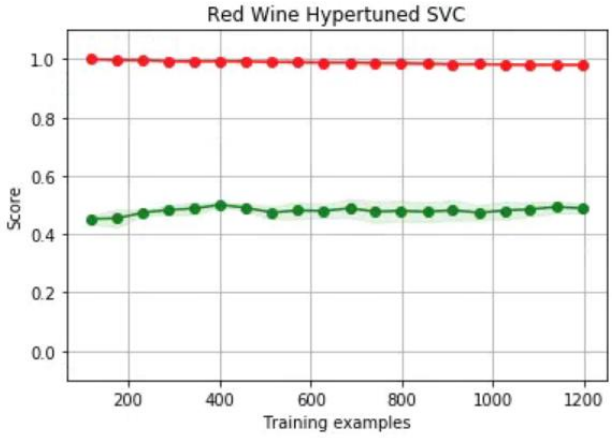

Fig. 12. Red wine support vector machine learning curve

TABLE XII. SUPPORT VECTOR MACHINE HYPERPARAMETERS

Model	C	Gamma	Kernel
Contraceptive use	2275.85	1.2e-3	rbf
Red wine	2.64	0.785	rbf

The two datasets both performed best with the RBF kernel but had very different C and gamma values. The contraceptive use model had a large C value corresponding to small boundary margin and small gamma value corresponding to a large standard deviation in the Gaussian (i.e. points further apart can still be considered similar.

The red wine model had a small(er) C value meaning a larger margin and a large(r) gamma meaning points had to be nearby to be similar. This may be due to the fact that the red wine dataset had more continuous quantitative variables that could be closer to each other in this distance metric.

Tables XIII and XIV show the confusion matrices. Similar to the other models, the contraceptive use is predicting well on the no use case (first row), moderately well on the short-term use case (third row) and fairly poorly on the long-term use case (third row).

TABLE XIII. SVM CONTRACEPTIVE USE CONFUSION MATRIX

84	6	29
25	19	29
27	9	67

TABLE XIV. SVM RED WINE CONFUSION MATRIX

0	1	0	0	0	0
0	1	6	6	0	0
1	1	99	28	2	0
0	1	35	88	6	1
0	0	2	19	18	0
0	0	1	1	3	0

The red wine confusion matrix again has almost all of the predictions on the diagonal (correct) or off-diagonal (misclassifying by one). The interesting outcome with this model is that it performs very poorly on the extreme ratings (first and last row) misclassifying all of them.

E. k-nearest neighbors

Using the *scikit-learn* KNeighborsClassifier I was able to develop models easily for my two datasets. The testing accuracy for the contraceptive use model was the worst of all the models we have used while the testing accuracy for the red wine dataset was second only to boosting (Table XV). Of note is that kNN has a fast training time and a relatively slower testing time as noted in the lecture videos.

Based on the results for the other models I am not surprised by the results of the contraceptive use model. There appears to be some overlap in the population between short-term use and no use, while long term use has been very unpredictable.

The results of the red wine kNN model were understandable if you consider that the majority of samples have very average scores and therefore they will cluster near each other, making them easier to classify.

The hyperparameters that were tuned are shown in Table XVI. Both models were optimized using a manhattan L1 distance metric and had a large number of neighbors in their models. The red wine model used a distance weight (giving a higher weight to those closer) which agrees with the previous statement that the average scores are clustering together.

Fig. 13 and 14 show the learning curves for these models. Similar to before the contraceptive use model shows a convergence in the training and testing accuracy while the red wine model shows an overfit training set but a testing set that shows improvement. More insight is need to understand what is going on in the red wine models that this keeps happening as all that I have now are some general observations and theories.

TABLE XV. K-NEAREST NEIGHBORS STATISTICS

Model	Training Accuracy	Testing Accuracy	Training Time	Testing Time
Contraceptive use	0.530	0.508	0.0103	0.0240
Red wine	0.686	0.656	0.0040	0.0225

TABLE XVI. K-NEAREST NEIGHBORS HYPERPARAMETERS

Model	Weights	N-neighbors	Distance Metric
Contraceptive use	Uniform	41	Manhattan
Red wine	Distance	85	Manhattan

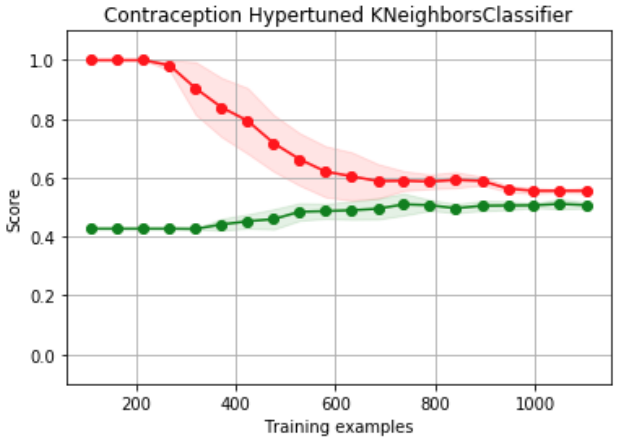

Fig. 13. Contraceptive use k-nearest neighbors learning curve

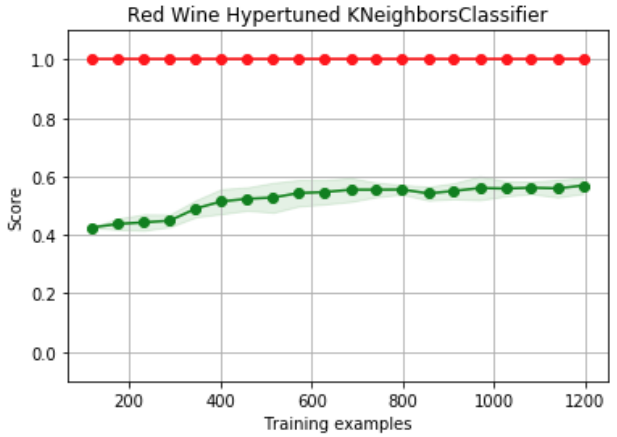

Fig. 14. Red wine k-nearest neighbors learning curve

Tables XVII and XVIII show the confusion matrices for the kNN models. While the contraceptive use matrix is very similar to previous models, the red wine matrix has all but two of the observations correctly classified (on the diagonal) or within one of being correct (on the off-diagonal)

TABLE XVII. KNN CONTRACEPTIVE USE CONFUSION MATRIX

67	17	35
26	26	21
28	18	57

TABLE XVIII. KNN RED WINE CONFUSION MATRIX

0	0	1	0	0	0
0	0	8	5	0	0
0	0	97	34	0	0
0	0	27	97	7	0
0	0	0	23	16	0
0	0	0	2	3	0

III. DISCUSSION

For the contraceptive use dataset, the best performing model was the pruned decision tree (Table XIX). It was also one of the fastest to train and the fastest to test. This data appears ideal for a decision tree because it can select the predictors that do not confound with each other in predicting very muddled classes. I was surprised that boosting did not improve this model. The kNN model was the worst performing since neighbors do not appear to be similar to one another in this hyperspace.

The red wine dataset performed best on the boosted decision tree, but this was followed closely by SVM and kNN. With kNN being the fastest to train and the decision tree being the fastest to test.

The neural network was one of the worst performers on both sets, in accuracy and time. It seems to require a large amount of domain knowledge to be useful, so hopefully this can be improved upon.

TABLE XIX. CONTRACEPTIVE USE ALL MODEL STATISTICS

Model	Training Accuracy	Testing Accuracy	Training Time	Testing Time
Decision Tree	0.556	0.580	0.0105	0.0025
Neural Network	0.561	0.536	11.2811	0.1707
Boosting	0.559	0.573	0.0710	0.0035
SVM	0.555	0.576	0.1178	0.0097
kNN	0.530	0.508	0.0103	0.0240

TABLE XX. RED WINE ALL MODEL STATISTICS

Model	Training Accuracy	Testing Accuracy	Training Time	Testing Time
Decision Tree	0.557	0.597	0.0150	0.0012
Neural Network	0.613	0.553	26.2831	0.7943
Boosting	0.693	0.659	1.3539	0.0210
SVM	0.651	0.644	0.1272	0.0160
kNN	0.686	0.656	0.0040	0.0225