CS7641: Analysis and Report for Assignment 1- Supervised Learning

Mingming Zhang

mzhang607@gatech.edu

Description of classification problems and experiments

In this assignment, I selected two datasets. One is the MINIST dataset, which includes the handwriting digits (One can download data from here *http://yann.lecun.com/exdb/mnist/*). The image data includes 10,000 different examples in total, each image represents one of the *10* handwriting digits from *0* to *9*, the data also contains the correct human label to indicate which number each handwriting digit is. The second dataset that I select is white wine quality dataset (One can download data here *https://archive.ics.uci.edu/ml/datasets/Wine+Quality*). It contains a total of 4898 samples and each observation descries the physicochemical and sensory variables of a wine, such as it is citric acid, residual sugar level and etc, and its related quality score rating which is a discrete number between 0 and 9 with higher score indicating a better wine.

I personally think these two datasets are interesting from the following two perspectives. On one hand, these two classification problems are useful in daily life. For example, when using mobile banking apps to deposit a check, it is important to for the app recognize the handwriting digits on a check to deposit accurate amount of money. When I do my grocery shopping, I would like to have some recommendation app to help me selecting a good wine using all the information that is available on the wine labels. One the other hand, these two datasets are with very different distributions, one is balanced dataset and the other one is unbalanced dataset (Figure 1). It will be interesting, from the machine learning perspective, to see how each dataset performs when conducting the experiments with different algorithms.

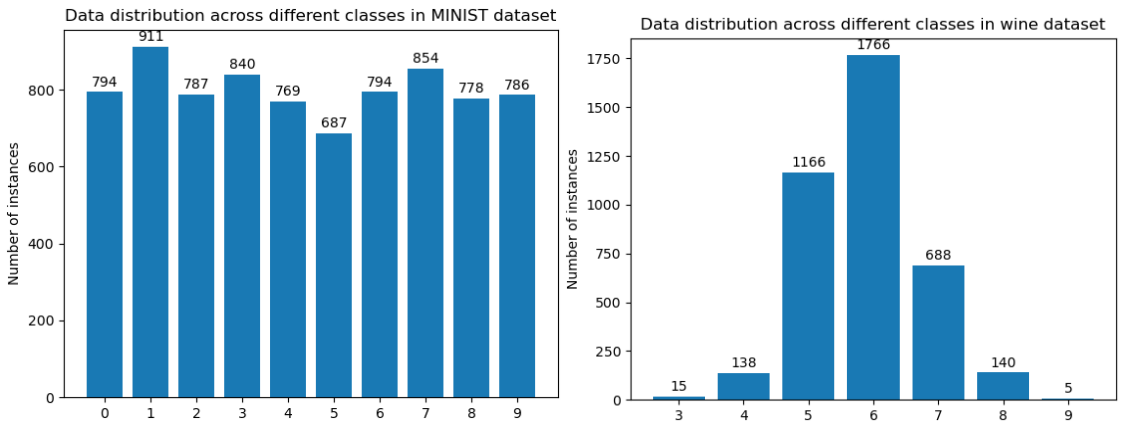

Figure 1: Distribution of number of instances across all the classes in two of the training datasets. Left subplot shows the MINIST training dataset is somewhat well balanced and number of instances varying between 687 to 911 in each class. **Right** subplot shows the wine training dataset is unbalanced, with quality score *6* has 1766 instances, while quality score *3* and *9* has only 15 and 5 instances, respectively.

For all the experiments in this assignment, I split each data set with 80% of it as the training data and 20% of it as testing data. After the split, the image dataset is with 8000 training examples and 2000 test examples; the white wine quality data is with 3918 training examples and 980 test examples. Figure 1 shows the details of the distribution of all the instances over all the classes. It shows that MINIST image data set is very balanced, while wine quality data is unbalanced.

In the experiments below of looking at the effect of using different sizes of data to train a model, I used 30%, 40%, 60%, 80%, 100% of training data, respectively, with 100% of the training data as all the training data being used. The data was used as it is, without any preprocessing or normalization. All the tests were conducted using the testing data mentioned above.

For each experiment ran on a given dataset, I arbitrarily select several hypermeters to tune. The hyperparameters were tuned by conduct 5-folder cross validation for each given set of hyperparameters. Then, a grid search was conducted in the hyperparameters' space to find the optimal solution using the accuracy as the evaluation criteria, unless stated otherwise.

Results

Decision Tree

During the experiments that I ran using Decision Tree (DT) algorithm, the number of hyperparameters that are used to tune and the parameters' range is as following:

- criteria: 'gini' or 'entropy
- splitter: 'best' or 'random'
- max_depth: range(2,20,4)
- min_samples_leaf: range(1,10,2)
- min_samples_split: range(2,6,2)

After the grid search on hyperparameters tuning, the best combination of the hyperparameters for the image dataset and wine quality dataset is shown in Table 1. Here we use the parameters 'max_depth', 'min_samples_leaf' and 'min_sample_split' as a way to prune the tree, which control the maximum depth a node can go, the minimum number of samples required to be at a leaf node, and the minimum number of samples required to split an internal node, respectively.

Table 1: best parameters selected from Decision Tree hyperparameter tuning experiments

Parameter	MINIST image dataset	Wine quality dataset
criteria	entropy	gini
max_depth	12	16
min_samples_leaf	1	1
min_samples_split	4	2
splitter	best	best

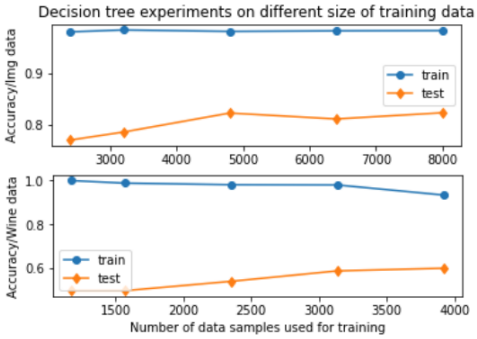

Figure 2: Experiments with Decision Tree models trained on different sizes of datasets. The model parameters used here are from the best parameters got from hyperparameter tuning experiment. Top subplot shows the results of the experiments conduct on Image dataset and bottom subplot shows the results from the experiments conducted on wine quality dataset.

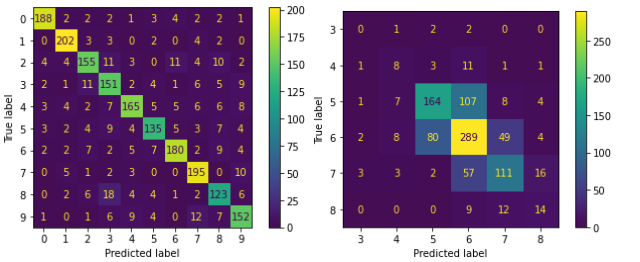

Figure 3: Confusion matrix of testing results when using Decision Tree algorithm. Left, confusion matrix of the testing results for MINITST data. Right, confusion matrix of testing results for wine quality data.

Figure 2 shows the results of the experiments on these two datasets. We can see that as the number data sets used for training is increasing, the better the testing accuracy will be. Overall, classifying MINIST data resulted in a better result as the testing accuracy can reach above 80% after using all the data available in training set, while using wine dataset the testing accuracy can only reach to 60%. The two experiments show that the models on these two datasets are general underfitting, and it might be possible to further improve the performance by feeding more data into the experiments. Figure 3 indicates the overall prediction results for each class for both datasets. From this figure, we can tell that, in the wine dataset, dominant classes with wine rating '5', '6' and '7' have been miss-classified a lot which determines the overall poor performance when classifying the wine quality dataset.

Neural Networks

During the experiments using Neural Networks model, the model was designed with three hidden layers with nodes 30, 50, 30, respectively. I used the 'adam' solver. The hyperparameters that I used to tune are as following:

- activation: 'relu' or 'tanh
- learning_rate_init: 0.01, 0.001, 0.0001
- shuffle: True, False
- alpha: 0.01, 0.001, 0.0001

Table 2: best parameters selected from Neural Networks hyperparameter tuning experiments		
Parameter	**MINIST image dataset**	**Wine quality dataset**
activation	relu	tanh
alpha	0.001	0.001
learning_rate_init	0.001	0.001
shuffle	False	False

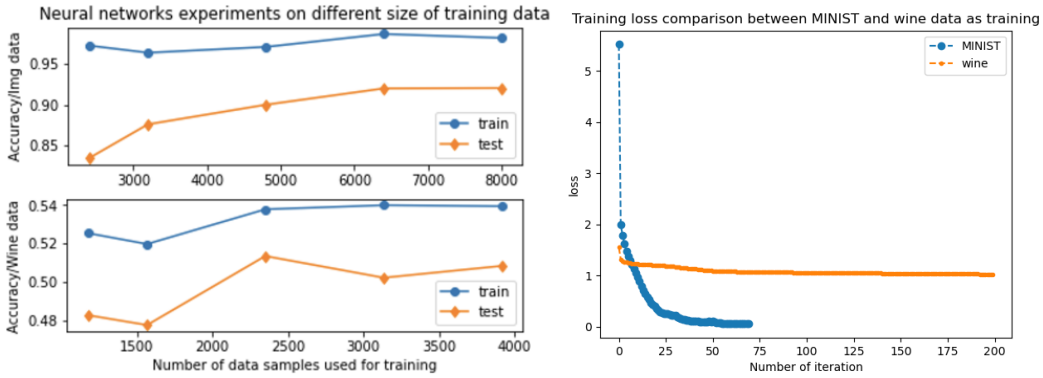

Figure 4: (Left) Experiments with Neural Network models trained on different sizes of datasets. The model parameters used here are from the best parameters got from hyperparameter tuning experiment. Top subplot shows the results of the experiments conduct on image dataset and bottom subplot shows the results from the experiments conducted on wine quality dataset. **(Right) training loss over the number of iterations when using the best hyperparameters to train with all the MINIST and wine data, respectively.** The experiment was using the best parameters mentioned in the Neural Network section. All the training samples were used in both experiments with the default settings for maximum number of iterations set at 200.

All the training data were used to tune the hyperparameter and the best selected hyperparameters are shown in Table 2. Then, the experiments using Neural Networks, with these best hyperparameters, were trained on different sizes of data and it showed that as increasing the size of the training data, testing performance will be increasing in general (Fig.4). Particularly, using MINIST dataset, it has a very good classification performance with testing accuracy reached above 90% when using all the

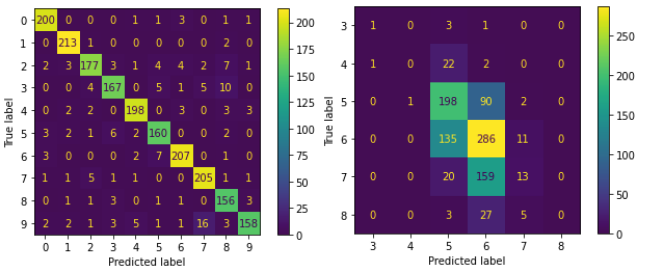

Figure 5: Confusion matrix of testing results when using Neural Network algorithm. Left, confusion matrix of the testing results for MINITST data. Right, confusion matrix of testing results for wine quality data.

training samples. Using wine dataset, the overall performance is not better compared to using Decision Tree algorithm before. The best test accuracy is around 50% (Fig. 4). When training size is higher than about 2400 samples, the performance starts to show the sign of overfitting, as increasing more training data starts to cause slightly decreasing in testing accuracy. Similarly, Figure 5 shows the performance is poor when classifying wine dataset because there a lot of miss-classified dominant classes with wine rating of '5','6','7' in this unbalanced dataset. The effect of balanced and extremely unbalanced data on the training process can also be viewed in Fig4, where using MINIST data the training loss is much lower and converge much quicker compared using wine dataset.

Boosting

During the experiment when using boosting algorithm (GradientBoostingClassifier), I didn't do a grid search, but used the following parameters. I used the following parameters as shown in Table 3. Here, I used 'max_depth' parameter to control the maximum depth each node in a tree can go as a way of pruning the tree.

Table 3: parameters used in experiments using boosting algorithm

Parameter	MINIST image dataset	Wine quality dataset
n_estimators	60	60
learning_rate	0.1	0.1
max_depth	8	8

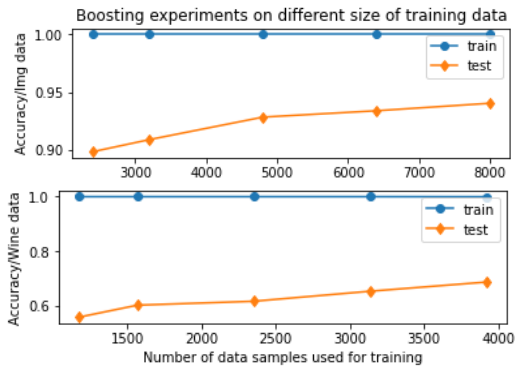

Figure 6: Experiments with Boosting models trained on different sizes of datasets. The model parameters used here are from the best parameters got from Boosting hyperparameter tuning experiment. Top subplot shows the results of the experiments conduct on Image dataset and bottom subplot shows the results from the experiments conducted on wine quality dataset.

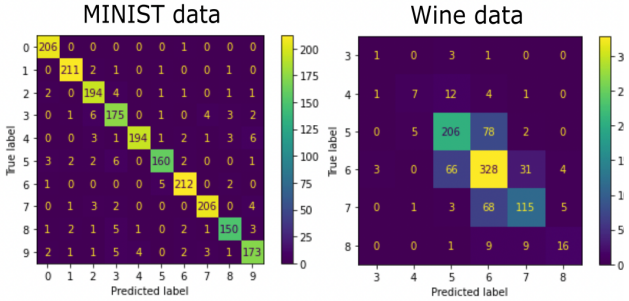

Figure 7: Confusion matrix of testing results when using boosting algorithm. Left, confusion matrix of the testing results for MINITST data. Right, confusion matrix of testing results for wine quality data.

During this experiment, we can see that more training data generally helps the trained model to achieve better performance when predicting the testing data. When using MINIST data as the training data, boosting algorithm-based model perform well with the testing accuracy reaching

around 94% when using all the training samples (Fig.6). When using wine quality data, the testing accuracy is much worse compared to when using MINIST data set, which is similar than other experiments mentioned before. However, the testing accuracy reach 69% when using all the wine training dataset (Fig.6). This is much higher compared to that when using Decision Tree and Neural Networks algorithm on wine quality data. Fig.7 shows the confusion matrix of the prediction results on the testing data. We can see that MINIST testing data has been well classified, while the wine testing data are dominantly misclassified by classes with rating score '5', '6', '7' which are primarily responsible for the much lower prediction accuracy when using wine quality data.

SVM

The hyperparameters that I used to tune during the SVM experiment are as following:

- kernel: 'rbf' , 'sigmoid', 'linear'
- C: 0.1, 1, 10
- gamma: 0.1, 0.3, 0.5

The best hyperparameters selected from the experiments are shown in Table 4.

Table 4: best parameters selected from SVM hyperparameter tuning experiments

Parameter	MINIST image dataset	Wine quality dataset
kernel	linear	rbf
C	0.1	1.0
gamma	0.1	0.5

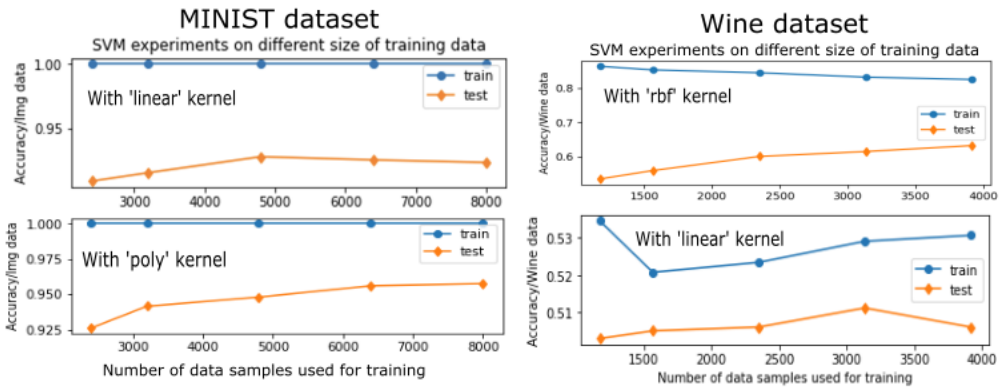

Figure 8: Experiments with SVM models trained on different sizes of datasets. In the top subplots for both MINIST and wine data, the model parameters used here are from the best parameters got from SVM hyperparameter tuning experiment. When using MINIST dataset, the linear model was used. The bottom plot shows the results using different kernel while keep other parameter unchanged.

The experiment results shown in Fig.8 indicates that increasing size of training dataset is generally helpful. It helped a little bit for MINIST data, as the results using 30% of all the training dataset it

already reached a very high testing accuracy level at 90.9% (Fig.8 top left subplot with 'linear' kernel). It helped more when using wine dataset, as the test accuracy increased from 50.8% to 62.8% as increasing dataset size from 30% to 100% of all available training dataset (Fig.8 top right subplot with 'rbf' kernel). From the confusion matrix of the testing experiment, I observed similar phenomena than before. SVM based algorithm performs general well when classifying MINIST data. When using wine quality data, the performance of predicting class with wine rating '5','6','7' mainly affect the overall poor performance of classifying wine quality dataset (Fig.9).

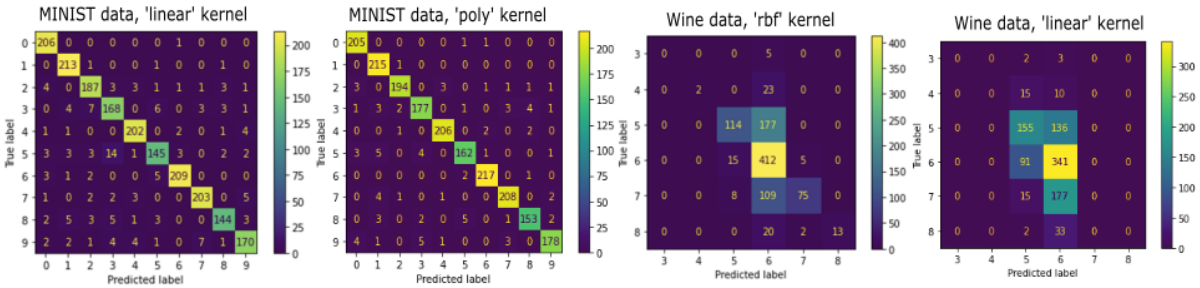

Figure 9: Confusion matrix of testing results. Left, confusion matrix of the testing results for MINITST data. Right, confusion matrix of testing results for wine quality data.

I also tested other two kernels, 'rbf' and 'sigmoid', on MINIST data set (results not shown here). Surprisingly, both training and testing accuracy is very low at around 10%. When I used 'poly' kernel, the testing accuracy reached 95% when all the training examples were used), which is even better than 'linear' kernel (Fig.8 bottom left subplot). For wine dataset, I also showed the results from using 'linear' kernel. The result was not as good as when using 'rbf' kernel (Fig. 8 bottom right subplot and Fig.9 4th subplot).

KNN

When conducting experiments using KNN based models, I changed the number of neighbors, K=3,6,10. As observed similarly compared to results showed before, using MINIST image data, the experiments has very good results with testing accuracy is higher than 90%. As training size going up, testing accuracy increased to around 95% (Fig.10). One observation is that the three values of 'K' I used here all causing similar testing accuracy around 94% when using all 8000 training examples. The performance of using wine quality data is general underfitting and has much worse

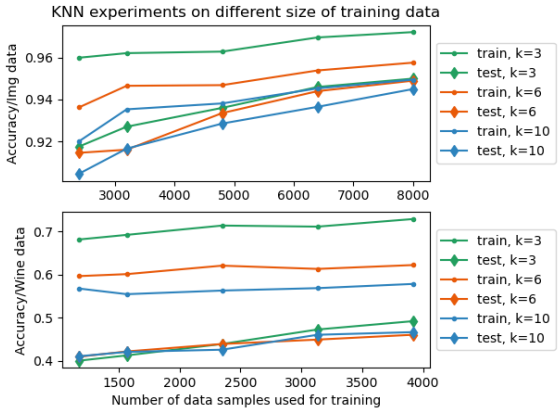

Figure 10: Experiments with KNN models trained on different sizes of datasets. Top subplot shows the results of the experiments conduct on image dataset and bottom subplot shows the results from the experiments conducted on wine quality dataset.

performance compared to using MINIST image data. Next, I further explore more values for 'k' by using all the training datasets (Fig.11). We see that when K value is set to 1, it can cause obvious over fitting, as the training accuracy is 100%, but the testing accuracy is much worse than that (Fig. 11). As increasing k value, the overfitting problem has been eliminated. When K>5, there is not too much improvement in MINIST experiments. And in wine experiment, increasing k values higher could still help to reduce the overfitting problem a litttle. But when k value is higher than 15, both training and testing perform similarly to each other (Fig.11). As mentioned before, the performance for classifying wine dataset is not very good and the test accuracy is lower than 50%.

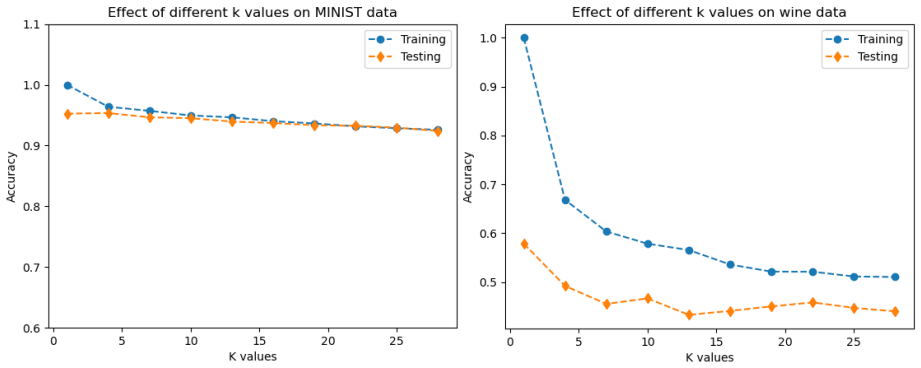

Figure 11: Experiments with KNN models trained on different K values. In this experiment, all the training data from MINIST and wine datasets are used to train the model, respectively. The testing is still conducted on the testing data.

Analysis and Discussion

In the experiments, MINIST dataset is well organized and with each class overall evenly distributed, while wine quality dataset is extremely unbalanced. The classes in wine data set with rating scores 3,4,8 and 9 that are with extremely low appearance frequency (Fig.1). Thus, it makes the MINIST dataset a very good classification problem overall, while the wine rating classification as a very challenging classification task. I used the testing accuracy to judge whether a model is performing good or not. When running all the experiments, boosting experiments were only done with one set of hyperparameters with 60 estimators due to its high computational expenses. During the experiment of using DT, Neural Networks and SVM, the best hyperparameters are selected by using the full datasets with 5-fold cross validation. KNN experiments were run with multiple k values.

The results obtained from running all the experiments show that classification for MINIST dataset usually ends up with a very good testing accuracy. Using DT algorithm, it has the worst performance in classifying MINIST dataset, with the testing accuracy is lowest at about 80% when used all the training data samples (Fig.12 bottom left). The experiment results showed that as training examples are increasing over 4800, the testing accuracy for DT model starts to decrease, showing a sign of overfitting. One of the reasons why decision tree performs the worst is probably

due the continuous nature of the features in the input image. The features are the values at each pixel, which are continuous numbers, instead of discrete categories values. This might cause decision tree-based algorithm can not perform well compared to other algorithm, such as SVM or MLP, because decision tree-based algorithm is maybe OK but not strongly capable to predicting continuous values.

When using Boosting, KNN, Neural Networks (MLP) and SVM algorithms, the decoding accuracies are very good and similarly ranging from 90 to 95%. In bottom left subplot of Fig.12, the result also shows that KNN algorithm has slightly better performance than others when using all the training samples to train the model, but the difference between KNN's performance and other algorithms' is not significant to conclude that KNN based model has the best performance. Overall, increasing training samples can increase the performance. However, when training examples are over 6400, we see testing accuracy for Neural Network model starts to decrease a little, showing the sign of over-fitting as more data is being used (Fig.12 left).

One interesting observation is that SVM based models works well with 'linear' and 'poly' kernels with default 3rd order when applied on MINIST data (Fig.8 left), while it performs terribly when using 'sigmoid' or 'rbf' kernel (results not shown due to the report length limit). One explanation is that MINIST dataset with pixel values as features are probably already well separatable by linear or lower order of polynomial boundary without doing complicated data transformation. But 'sigmoid' or 'rbf' would convert dataset into some more complicated representation in which it cannot be well separated by linear boundary anymore after transformation. While in wine dataset, 'rbf' kernel works much better compared to using linear kernel (Fig.8 right). It states that wine dataset needed more transformation to be linearly separatable during classification.

Speak of wine dataset, most of the algorithms tried here cannot perform very well. The best algorithm in predicting wine quality score is boosting and its highest predicting accuracy at 69% when using all the training data samples (Fig.12, right). SVM, DT and Neural Networks (MLP) are the next few ranked algorithm with predicting accuracy range from 50 to 60% when using all the training samples.

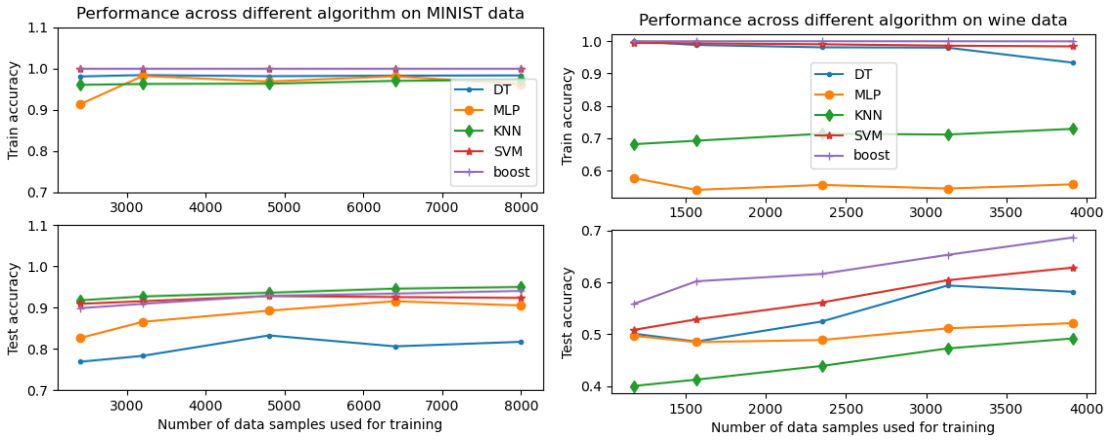

Figure 12: Comparison of the performance when using different algorithms together. Here, K=3 when used in KNN algorithms. Other algorithms are using the best parameters or parameters mentioned in the sections above.

The worst algorithm in predicting wine quality is KNN, with an accuracy around 49%. KNN experiments are done with three different k values in the initial experiments (Fig. 10). In addition, I also tested the performance of increasing K values when using all the training examples in both MINIST data and wine dataset (Fig.11). After K>=3, the performance of the model becomes less overfitting and more stable, especially for MINIST dataset. Thus, I used K=3 in KNN model when comparing the performance to other algorithms (Fig.12). One thing to notice is that as the number of training examples increase in wine dataset, the testing performance is increasing for all the algorithms. In the case of using DT algorithm, when training examples are bigger than 3200, it starts to show decease in performance as increasing training samples (Fig.12 bottom right subplot). The overall poorer performance when using wine dataset across various algorithms indicates that the root cause of the poor performance might be the severe imbalance in wine data set itself, instead of what algorithm was used.

Because of this, in a different experiment I also tried standardizing the wine quality data (results not shown here). In the wine dataset, different attributes, such as 'fixed acidity', 'residual sugar', 'pH', 'alcohol' and etc, are with very different numbers because their measured units are different. Usually, standardizing the attributes before feeding them into the model to train might help to minimize the bias caused by different types of measurements. Unfortunately, the standardized dataset did not help to improve the performance significantly, but with testing accuracy that might be 1-2% higher across all the experiments compared to the results when not using standardization on the datasets. Again, standardization of the wine dataset is not helping the poor testing performance across all the algorithms, and I think the reason is probably due to the dataset is severely imbalanced.

Table 5: time cost for training and testing when using all the data samples (time in s).

	MINSIT dataset		**Wine quality dataset**	
	Train	**Test**	**Train**	**Test**
Decision Tree	1.1669	0.0008	0.0275	0.0005
Neural Networks	3.9575	0.0034	4.8176	0.0032
Boosting	338.27094	0.04178	6.17942	0.01654
SVM	3.30185	0.79287	1.93488	0.36857
KNN	0.00749	0.60086	0.00356	0.02240

Another observation to indicate the poor performance on wine dataset is the loss curve as increasing number of iterations during Neural Networks experiments (Fig.4, right). As the number of iterations going up during Neural Network training, the experiment on MINIST dataset is with loss that is decreasing and finally converged at 70^{th} iteration. In contrast, the loss curve of using wine quality dataset is running up to the defaulted 200 iterations without the 'Adam' solver converged. This is the reason why training on a larger MINIST dataset is only taking 3.9575s, while it takes a longer time of 4.8176s to train on the smaller wine quality dataset (Table 5). In addition, the hyperparameters used in Neural Networks experiments are obtained from the best hyperparameters selected from the grid search with 5-fold cross validation experiment, where a wide range of parameters were testing, such as learning rate was tested within a range from 0.001 to 0.01, L2 penalty rate was tested within a range from 0.0001 to 0.01. The best

hyperparameters still perform poorly on wine dataset prediction with the testing accuracy is at only around 50%. These experiment results could indicate that simply using wine quality data as what it is with more iterations or more data sample to train will not help increase the performance because the wine quality dataset is severely imbalanced and directly predicting rating score might not be a good classification problem.

The time cost for training models and testing using each algorithm shown in Table 5 also indicate another interesting observation, that KNN is taking much less time to train but it takes longer time than Neural Networks and Boosting models to predict. This is because KNN is simply remembering all the input samples during training and comparing the given new examples to its remembered training samples during testing. As to other algorithms, they usually take much longer during training phase as they need to update their weights and working towards to minimize the training loss, while during testing they can simply use the weights they have already trained. As a result, training is taking a much longer time compared to testing, when using DT, Neural Networks, Boosting and SVM algorithms.

In all the experiments, boosting algorithm is taking a very long time to train compared to using other algorithms, especially when using image dataset. This is because the 'GradientBoostingClassifier' repetitively building a tree in an improving or boosting way to conduct predictions to finish one round of training. For the instances here, the number of estimators is 60, that means this method will repeat the processing of building the tree over the given datasets for 60 times to minimize the loss. This is the main reason why it takes so long to finish the training task. It could also explain why boosting learners are achieving one of the best results shown in the experiments above, especially when using wine quality data (Fig.12 bottom right subplot), because it starts with the given dataset to train an initial estimator and then assigning misclassified examples with more weights into the following round of estimator building. This boosting process allows the estimators to learn the misclassified examples in a recursive way, which helps refining the classification boundary to fit the training data better. This boosting process is also generally robust for overfitting; thus, it makes the model generalization much better.

In conclusion, MINIST dataset itself is well balanced across different classes and each attribute (the image pixels) are already kind of normalized because it's based on image pixel values. On the other hand, wine dataset is with severe imbalanced data problem. I am thinking of a different approach to improve predicting wine quality data, which is to treat the wine dataset differently, rephrasing it into a different classification problem. Instead of classifying the discrete wine rating scores directly, we can segregate the rating scores into several new sub-classes. For example, any wine with rating score lower than 6 is considered as 'bad' wine, rating with score 6 is considered as 'OK' wine, rating with score higher than 6 is considered as 'good' wine. As a result, the transformed dataset is much more balanced and thus make it a much more suitable dataset for classification.