Supervised Learning

Supervised Learning is one of the most important class of techniques in machine learning. By using certain learning algorithm, a model can be built from data, and used for prediction. A common practise is to train the model using first part of a dataset, which is called the training set, and the performance of the model is evaluated by comparing the forecast with the actual value in the remaining data, which is called the testing set.

In this project, two datasets are used to compare the performance of five supervised learning algorithm, and to show the effect of the parameters setting of each algorithm. The first 60% of the dataset is used as the training set, and the remaining 40% are used as the testing set.

It will be interesting to see how much data is needed for the algorithm to work well, and thus for each of the five algorithms, the accuracy rate of in-the-sample and out-of-sample prediction of various size of dataset (from a small dataset till the whole training set) are calculated (the learning curve). The in-the-sample accuracy are simply estimated by using the same data for learning and prediction, while the out-of-sample prediction accuracy is estimated using 10-fold cross validation.

In addition, in each of the five algorithms, there are some parameters which affect the performance of the model. Therefore, the out-of-sample prediction accuracy are evaluated using different the parameters, and each model is then calibrated by 10-fold cross validation using the training set.

At the end, the out-of-sample performance, e.g. accuracy, speed, of each of the five algorithms are compared among using the testing set. Since the two datasets have different properties, the results can be different.

Dataset 1 – Default of Credit Card Client

The first dataset is related to the default of credit card clients of customersâ€™ default payments in Taiwan. It will be interesting for banks to know which kind of clients are more likely to default their credit card payment and thus have a better business strategy.

This dataset includes the record of payment history of 30000 clients. The dependent variable is if the client obligated the payment or not, and there are 23 independent variables:

X1: Amount of the given credit including both the individual consumer credit and his/her family (supplementary) credit.
X2: Gender (1 = male; 2 = female).
X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
X4: Marital status (1 = married; 2 = single; 3 = others).
X5: Age (year).
X6 - X11: History of past payment.
X12 - X17: Amount of bill statement.
X18 - X23: Amount of previous payment.

In this project, only X1 to X11 is considered for modelling for sake of simplicity and the purpose of avoiding cruse of dimensionality.

Source: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

Dataset 2 – Equity Return Prediction under Fama-French Three Factor Model

The second dataset is related to prediction of equity return under Fama-French Three Factor Model. It is interesting to see if equity return can be predicted by some factors, instead of being some white noise. This is essential in testing if the market is efficient under efficient market hypothesis, and also it may lead to profitable trading strategies.

The linear form of the model considered is: $ER_t = \beta_0 + \beta_1 ER_{t-1} + \beta_2 SMB_t + \beta_3 HML_t$

where t denotes the time, and the sampling frequency is daily,
ER is the excess equity return (equity market return – risk-free rate),
SMB is return of Small [market capitalization] Minus Big,
HML is return of High [book-to-market ratio] Minus Low.

Since this project is about classification, the dependent variable is $1\{ER_t \geq 0\}$ (1 if the excess return is positive or zero, and 0 otherwise), i.e. it is to predict the market is going up or going down by the three variables. In addition, only data on of after 19700102 is used, and thus totally 11142 days of data is available.

One more point worth mentioning is that this is a time series dataset, which means that order matters. One potential issue here about such a time series dataset is that there are un-modelled autocorrelation (other than the AR(1) that is already include in the linear form), and there are potential structure change as time passes. The aim of this analysis is trying the find of the pattern in long run instead of any short-term structure.

Source: http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

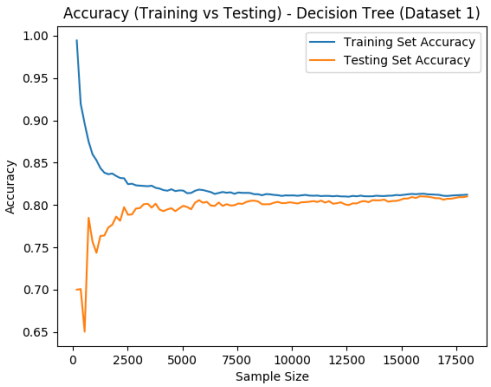
Decision Tree

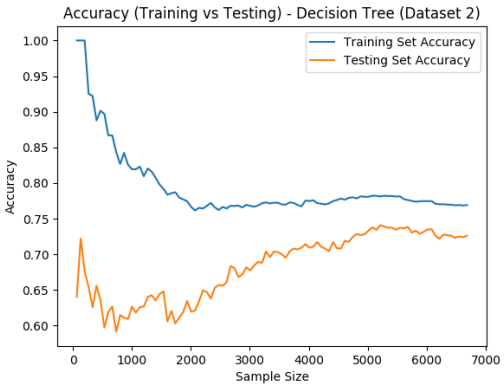
A decision tree is used. Gini index, which is a criterion to minimize the probability of misclassification, is used for deciding which variable should be used for classification.

Learning Curve:

Testing Accuracy: for both Dataset 1 and 2, the in-the-sample accuracy is high when the sample size is really low, and then gradually decreases and stabilize when the sample size become larger. Most likely it is because it is much easy for a model to fit the data when there are just a few data point, and it became more difficult when there are more.

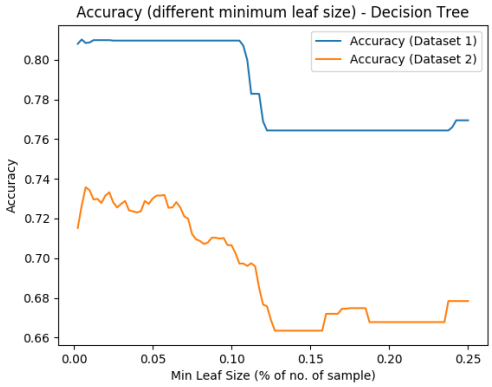
Testing Accuracy: for both Dataset 1 and 2, the out-of-sample accuracy is increasing as the sample increase. The reason is simple: a sufficient amount of information is needed to approximate the relationship, and thus give a better estimation.





Parameter Calibration:

One key parameter of the decision tree is size of the whole tree. It can be either controlled by limited the height of the tree, or the size of the leaf. An out-of-sample prediction (by 10-fold cross-validation, using the training set) accuracy of across different minimum leaf size (as % of number of data point) is shown. The leaf size that produce the most accurate result are chosen. (Dataset 1: 0.5%, Dataset 2: 0.75%)



In this case, the typical case of overfitting due to too small leaf size is not really that clear, but underfitting due to too large leaf size is obvious – the accuracy is lower when the leaf size is too large.

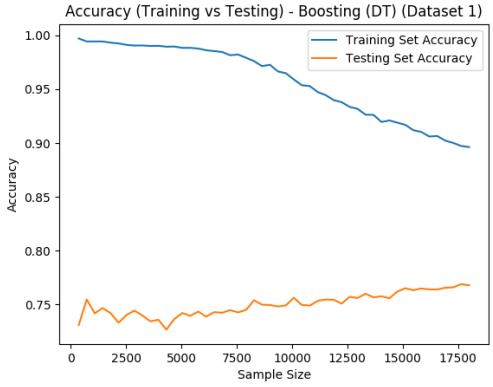
Boosting

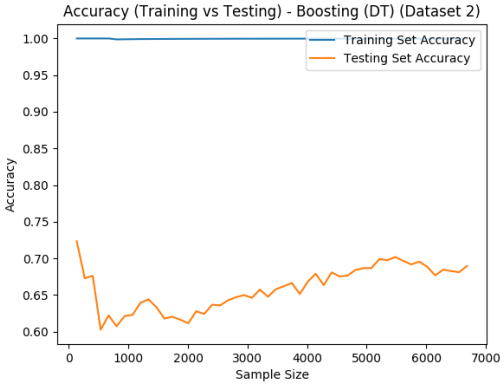
Adaptive Boosting of the previous decision tree is used. The aim of boosting is trying to improve the performance by focusing more on fitting the more difficult data point.

Learning Curve:

Testing Accuracy: when compared to the decision tree case, the in-the-sample accuracy is higher for both dataset.

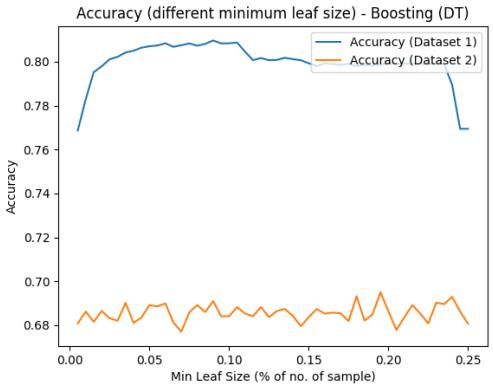
Testing Accuracy: boosting produce a similar out-of-sample accuracy as the decision tree case when the sample size is small, but it is way better when the sample size is relatively small.





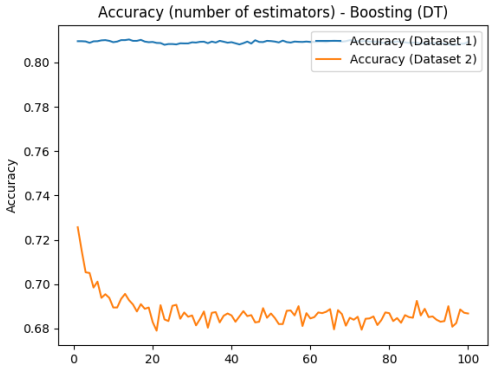
Parameter Calibration:

Same as a typical decision tree, the leaf size matters. The leaf size that produce the best accuracy rate will be chosen. (Dataset 1: 9.0%, Dataset 2: 19.5%)



It is clear that there is some kind of overfitting in the Dataset 1 if the leaf size is too small, and thus a larger leaf size, i.e. more aggressive pruning, is needed.

This also shows the downside of boosting is: it may lead to easier overfitting in some cases – this is clearer in the Dataset 2. The most accurate out-of-sample fit is obtained by using only 1 estimator in boosting, i.e. no boosting. (Dataset 1: 14 estimators, Dataset 2: 1)



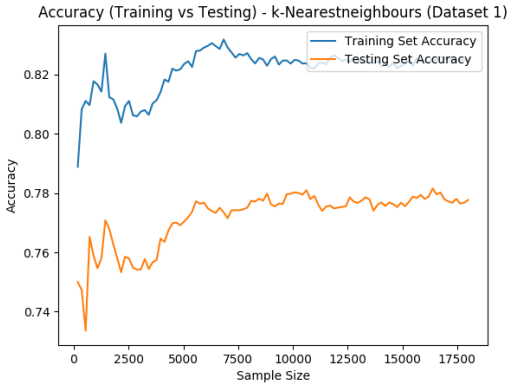
k-Nearest Neighbours

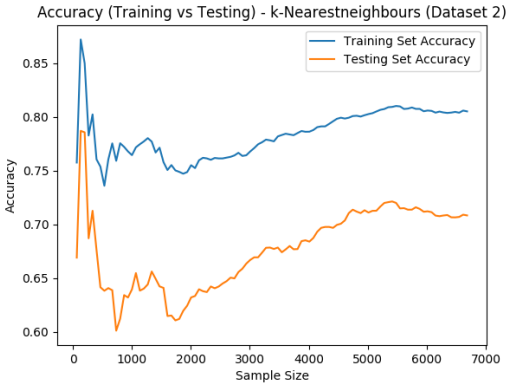
A k-Nearest Neighbours algorithm is used.

Learning Curve:

Dataset 1: Both the in-the-sample and out-of-sample accuracy are generally increasing as there are more data, and stabilize afterwards.

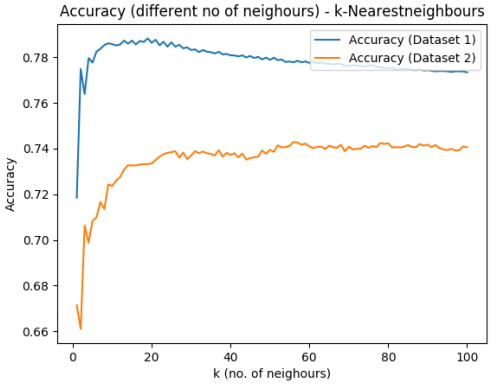
Dataset 2: Both the in-the-sample and out-of-sample accuracy are at a higher level at the first when only a short time series is used, and the decrease to a lower level sharply, and then increasing as there are more data, and stabilize afterwards.





Parameter Calibration:

One key parameter of the k-Nearest Neighbour algorithm is the k, which is the number of neighbours considered. An out-of-sample prediction (by 10-fold cross-validation, using the training set) accuracy of across different minimum leaf size (as % of number of data point) is shown. The parameter k that produce the most accurate result are chosen. (Dataset 1: 19, Dataset 2: 56)



In this case, the typical case of overfitting due to too small leaf size is not really that clear, but underfitting due to too large leaf size is obvious – the accuracy is lower when the leaf size is too large.

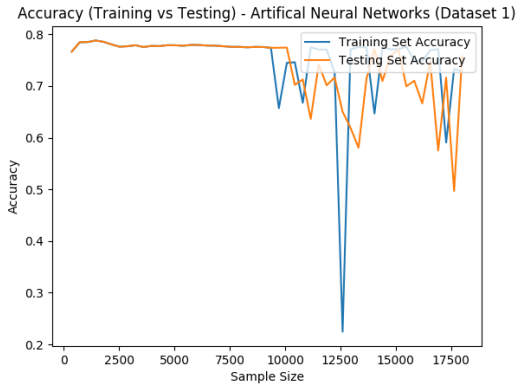
Neural Networks

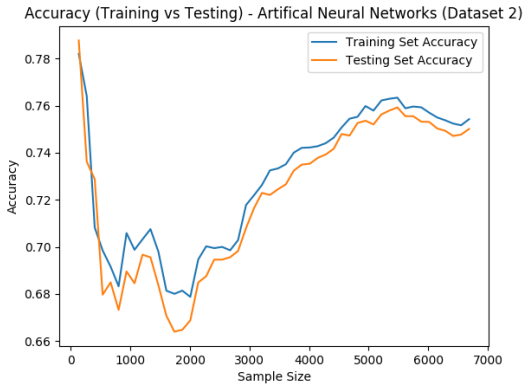
An artificial neural network with totally three layers, i.e. only 1 hidden layer, is used.

Learning Curve:

Dataset 1: Both the in-the-sample and the out-of-sample accuracy are at a high level except one case, which is likely due to computational issue.

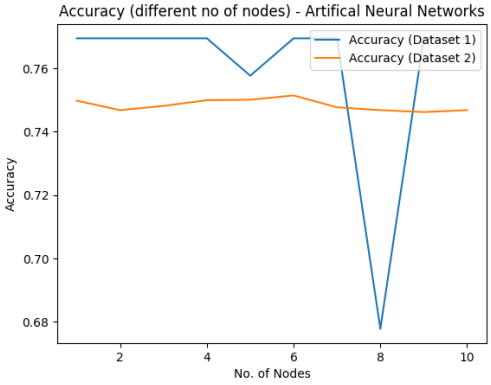
Dataset 2: Both the in-the-sample and the out-of-sample accuracy are high when the sample size is really low, and then it drops sharply, and then gradually increase and stabilize when the sample size become larger. Most likely it is because it is much easy for a model to fit the data when there are just a few data point, and it became more difficult when there are more.





No. of perceptron in the hidden layer:

The number of perceptron in the hidden layer is yet to be determined. An out-of-sample prediction (by 10-fold cross-validation, using the training set) accuracy of across different number of perceptron is shown. The number of perceptron that produce the most accurate result are chosen. (Dataset 1: 1, Dataset 2: 6)



However, for both Dataset 1 and 2, it seems that different number of perceptron return really similar results in most cases. It somehow indicates that this 1-layer neural network maybe not be ideal for these dataset.

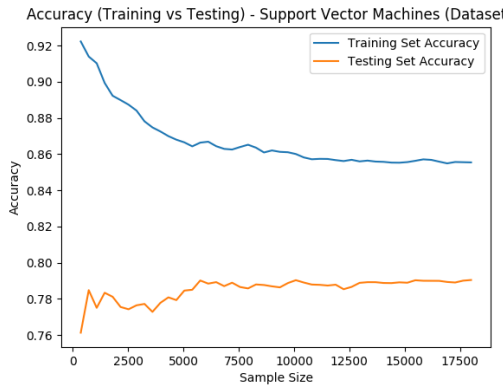
Support Vector Machines

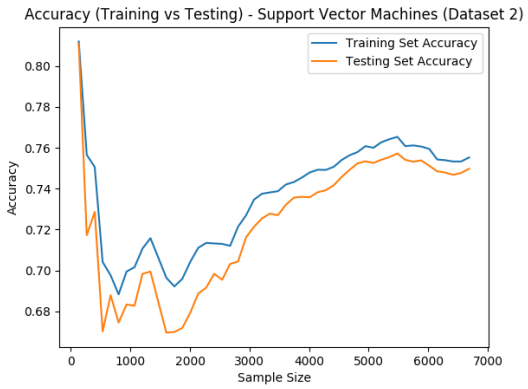
A support vector machine is used.

Learning Curve:

Dataset 1: The in-the-sample accuracy is generally decreasing as there are more data, and stabilize afterwards, most likely because of the really good fit when there is only fit data. The out-of-sample accuracy is basically unchanged when there are more and more data, showing that the support vector machine model doesn't require too many data to learn.

Dataset 2: Both the in-the-sample and the out-of-sample accuracy are high when the sample size is really low, and then it drops sharply, and then gradually increase and stabilize when the sample size become larger. Most likely it is because it is much easy for a model to fit the data when there are just a few data point, and it became more difficult when there are more.

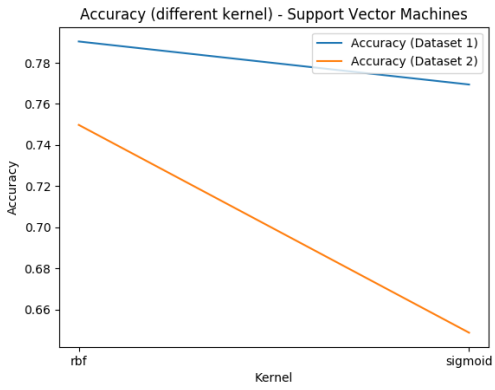




Choice of Kernel:

The most common kernel. radial basis function, is used in the analysis. But it is no tthe only one available, and a sigmoid kernel is tested also to see if it provides a better out-of-sample fit.

However, the out-of-sample accuracy (10-fold cross validation on the training set) shows that the radial basis function is a better kernel for both Dataset 1 and 2.



Comparison

The performances of the five algorithms for Dataset 1 and 2 are then compared.

In-the-sample Accuracy Rate	Dataset 1	Dataset 2
Decision Tree	81.22%	76.92%
Boosting	81.11%	75.29%
k-Nearest Neighbours	79.61%	76.14%
Neural Networks	76.94%	75.09%
Support Vector Machines	85.54%	75.53%

The in-the-sample accuracy is similar among all methods for Dataset 2, while the decision tree classier is marginally the best among all five. For Dataset 1, SVM is the best among all five.

Out-of-sample Accuracy Rate	Dataset 1	Dataset 2
Decision Tree	83.64%	52.50%
Boosting	83.67%	52.14%
k-Nearest Neighbours	81.18%	52.25%
Neural Networks	79.28%	52.17%
Support Vector Machines	81.57%	52.43%

One weird finding for out-of-sample accuracy for Dataset 1 is: the out-of-sample accuracy rate is slightly higher that that of the in-the-sample one, which is yet to investigate. The accuracy rate is more than 80% in most cases, showing that it is not too difficult to estimate which clients is going to default their credit card given the personal information and payment history.

For Dataset 2, the out-of-sample accuracy rate is similar among all five methods, and are all just slightly above 50%, showing that it is quite hard to predict the market.

Time spent (sec.)	Dataset 1	Dataset 2
Decision Tree	0.13	0.04
Boosting	0.74	0.30
k-Nearest Neighbours	0.84	0.11
Neural Networks	2.27	2.97
Support Vector Machines	76.81	5.11

Obviously, SVM is the much more slower than other four methods, while decision tree is the faster one.

For both Dataset 1 and 2, I will suggest to use the decision tree with or without boosting for prediction as it has good out-of-sample accuracy forecasting, and at a high speed.