Name: Yongchao Tang
Email: yongchaotang@gmail.com

Questions and Report Structure

1. Classification vs Regression

Your goal is to identify students who might need early intervention - which type of supervised machine learning problem is this, classification or regression? Why?

Answer: The type of the problem is classification. The reason is that the groups of the students which we should classify are discrete values.

2. Exploring the Data

Can you find out the following facts about the dataset?

Total number of students

Answer: 395

Number of students who passed

Answer: 265

Number of students who failed

Answer: 130

Graduation rate of the class (%)

Answer: 67.09%

Number of features (excluding the label/ target column)

Answer: 30

Use the code block provided in the template to compute these values.

3. Preparing the Data

Execute the following steps to prepare the data for modelling, training and testing:

Identify feature and target columns
Preprocess feature columns
Split data into training and test sets
Starter code snippets for these steps have been provided in the template.

4. Training and Evaluating Models

Choose 3 supervised learning models that are available in scikit-learn, and appropriate for this problem. For each model:

Model: Support Vector Machines

What are the general applications of this model? What are its strengths and weaknesses?

Answer: Support vector machines are generally applied in text and hypertext categorization, image classification and hand-written characters recognition.

The advantages of SVM are:

First, it is effective for high dimensional phase spaces and cases where number of dimensions is larger than the number of samples.

Second, versatile kernel functions can be specified for the decision process. Thus the SVM is sufficiently applicable for different kinds of problems including classification and regression.

Third, the so-called support vectors only refer to a subset of training points, so it is memory efficient.

The disadvantages of SVM are:

First, it might give poor performances when the number of features is much greater than the number of samples.

Second, it does not provide probability estimates so it cannot be easily applied in the cases where a quantitive likelihood is required.

Given what you know about the data so far, why did you choose this model to apply?

Answer: There are totally 48 features including the dummy variables in the dataset, and the number of training sample is 300. So the ratio between feature size and training sample size is around 1/6. Because SVM is effective for high dimensional state spaces, I think I can choose SVM as one of the training models to analyze the data. Furthermore, different kernel functions can provide more tuning parameters and may lead to high accuracies.

Model: Nearest Neighbours

What are the general applications of this model? What are its strengths and weaknesses?

Answer: Nearest neighbours are a common method for machine learning and data analysis. It can be generally applied to numerous fields of problems including pattern recognition, data compression, recommendation systems, search, cluster analysis and so on. In the last project, I was suggested by a reviewer to use the K nearest neighbours method to verify the predicted price of a house calculated by the Decision Tree method.

The advantages of nearest neighbours are:

First, as an instance based learning method, it actually remembers the whole training data and nothing is thrown away.

Second, given a fixed dimension, a semi-definite positive norm, and n points in this space, the nearest neighbour of every point can be found in O(n*log(n)) time and the m nearest neighbours of every point can be found in O(m*n*log(n)) time. Thus it is fast to learn.

Third, the query time for the K nearest neighbours is O(log(n)+K), so it is fast for query.

The disadvantages of nearest neighbours are:

First, the space for the story of nearest neighbours is O(n), which is as large as the sample size. For big data analysis, it costs much storage.

Second, it is easily to overfit by bottling the whole data.

Third, it is sensitive to noise data.

Given what you know about the data so far, why did you choose this model to apply?

Answer: I choose the method because it can be trained and queried quickly. Also, it can treat the data with discrete labels which are included in our dataset. Furthermore, it is easy to interpret the result by checking the nearest neighbours of the test data.

Model: Decision Trees

What are the general applications of this model? What are its strengths and weaknesses?

Answer: Decision Trees can be generally applied in control system, object recognition, financial analysis, text processing and so on.

The advantages of decision trees are:

First, trees are easy to interpret and can be visualized. A given problem is observable in a decision tree, thereby the explanation for the condition is easily explained by boolean logic.

Second, the query time if O(log(n)), n is the number of data points used to train the tree. So it is fast to predict.

Third, it is able to handle both numerical and categorical data.

Fourth, it is able to solve multi-output problems.

Fifth, it is possible to be validated by statistical tests. That makes it possible to account for the reliability of the model.

The disadvantages of decision trees are:

First, it is easy to overfit the training data.

Second, small variations in the data might result in a completely different tree being generated. So it is unstable.

Third, practical decision-tree learning

algorithms are based on heuristic algorithms such as the greedy algorithm. Consequently, global optimal decision tree cannot be guaranteed.

Fourth, some functions such as XOR, parity or multiplexer problems are difficult to be expressed with decision tree.

Fifth, some classes dominating the training dataset may lead to biased trees.

Given what you know about the data so far, why did you choose this model to apply?

Answer: One key factor in the building of a model is the interpretation of results. The decision tree are easily interpreted, thereby we can use the model to answer which factors account for the students' performances.

Fit this model to the training data, try to predict labels (for both training and test sets), and measure the F1 score. Repeat this process with different training set sizes (100, 200, 300), keeping test set constant. Produce a table showing training time, prediction time, F1 score on training set and F1 score on test set, for each training set size.
Note: You need to produce 3 such tables - one for each model.

Answer:

SVC	100	200	300
Training time (secs)	0.0013	0.0030	0.0064
Prediction time (secs)	0.0009	0.0012	0.0018

F1 score for training set	1.0	0.985185185185	0.971698113208
F1 score for test set	0.776315789474	0.776315789474	0.791946308725

KNeighbors Classifier	100	200	300
Training time (secs)	0.0007	0.0006	0.0005
Prediction time (secs)	0.0013	0.0023	0.0024
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.791666666667	0.753424657534	0.791666666667

DecisionTreeClassifier	100	200	300
Training time (secs)	0.0010	0.0013	0.0017
Prediction time (secs)	0.0003	0.0001	0.0001
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.655172413793	0.741935483871	0.753846153846

5. Choosing the Best Model

Based on the experiments you performed earlier, in 2-3 paragraphs explain to the board of supervisors what single model you choose as the best model. Which model has the best test F1 score and time efficiency? Which model is generally the most appropriate based on the available data, limited resources, cost, and performance? Please directly compare and contrast the numerical values recored to make your case.

Answer:

I coarsely tuned three models: SVC, K neighbours, decision tree. From the above table, the F1 scores for test set by training with the whole train data are 0.79 for support vector classifier, 0.79 for k neighbours classifier and 0.75 for decision tree classifier. Therefore, the SVC is chosen by me as the best model.

The K neighbours classifier gives the best time efficiency for training process, which takes 0.0005 sec for 300 training dataset. It also has the best test F1 score 0.7917, as good as that of SVC 0.7919. While the decision tree classifier gives the least prediction time 0.0001 secs on test set.

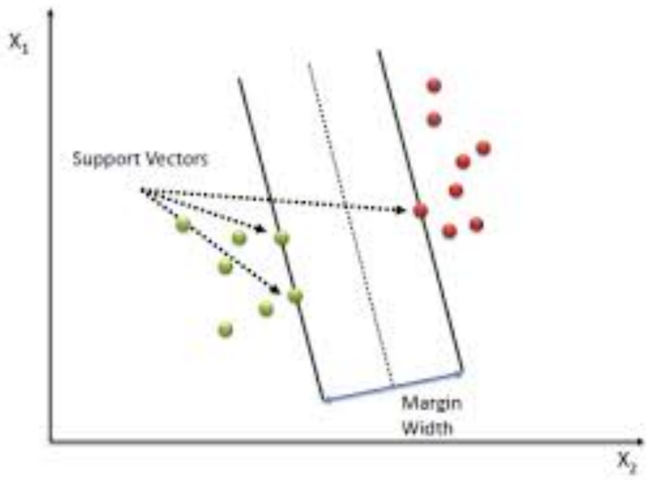
Based on the limited resources, cost, I choose SVC as the most proper model for the following reasons: First, the time cost may not be an issue due to limited training and test dataset. Second, the model give the best f1 score 0.7919 on test set after being trained with the whole training set. Its f1 score for test set grows and the f1 score for training set decreases as the training size grows, which complies with the curve line we have learnt.

In 1-3 paragraphs explain to the board of supervisors in layman's terms how the final model chosen is supposed to work (for example if you chose a decision tree or support vector machine, how does it learn to make a prediction).

%%%

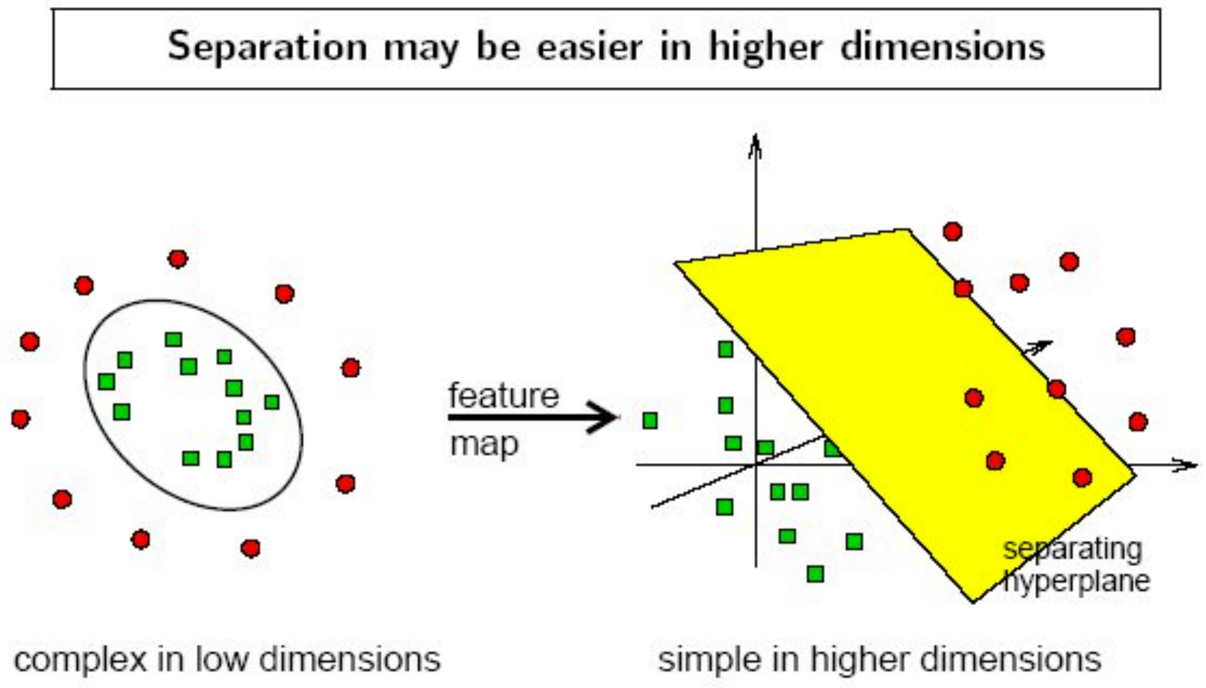
Answer: Support vector classifier is a learning model with associated learning algorithms that analyze training data used for classification. The training examples are a collection of data points. These points can be specified with vectors such as coordinate vectors (x1, x2) of positions. Each data point is also marked for belonging to one of two categories. Given the training examples, A support vector classifier can learn the training data to build a model that assigns new examples into one category or the other. These separate categories are divided by a clear gap that is as wide as possible. New examples are then predicted to belong to a category based on which side of the gap they fall on.

The learning process of the algorithm is to tune a set of parameters to maximize the distance between two lines and find the widest gap, also called margin width which divides the separate categories. As shown in the graph below, the two lines by which the green and red dots region are bounded can be expressed by a1*x1 + a2*x2 - b = 1 and a1*x1 + a2*x2 - b = -1, respectively. The a1, a2, b are all tuning parameters. The points (x1,x2) are training data points. By making the distance 2/sqrt(a1^2 + a2^2) the largest, we find a set of parameters that can distinguish the two groups of data. In the process, the data points on the boundaries are the most important for the determination of the gap. Therefore,



the data vectors associated with these data points are called support vectors.

During the optimization of the distance, the correlation between any two of the data points have to be calculated. The function that return this value is called kernel function. Typical kernel functions are dot product, the square of the dot product and so on. For example, by using the square of the dot product, two-dimension data points are actually mapped onto a three dimensions space. And separation may be easier in higher dimensions for the nonlinear case as shown in the left plot below.


Separation may be easier in higher dimensions
feature map
separating hyperplane
complex in low dimensions
simple in higher dimensions

It is clear that there is no line which can separate the green and red dots. After applying the kernel function, the square of the dot product on any pair of data points and mapping the data points onto three dimensions, a plane can be drawn to separate two groups of data. A plane in the three dimensions is a counterpart of a line in two dimensions. In the case of the state space with more than two dimensions, the counterpart of a line is called a hyperplane. In all, with the kernel trick, support vector classifier can use linear functions to

separate non-linear separable datasets by adding more information into any pair of data points. This priori experience for the data is the bias.
%%%

Fine-tune the model. Use gridsearch with at least one important parameter tuned and with at least 3 settings. Use the entire training set for this.

What is the model’s final F1 score

Answer:

Actually, I found that if I set the gamma:[0, 1], the grid search method will return the result in which gamma = 0, instead of the best parameter I found gamma = 0.1 for the test set. I think that the optimal parameter calculated for the training set may not be optimal for the test set. As I have found from the fluctuation of f1 score on training set with K neighbours classification, the training set may have unbalanced data. Therefore, the grid search method may not be effective for the unbalanced data.

With help of the reviewer, I used the stratified shuffle split data-split to get over the fact that the dataset is strongly imbalanced towards one of the two target labels. The stratified shuffle split the training data into groups with almost the same percentage of positive over negative as the others, thereby reducing the harmful effect of the unbalanced data structure. As a result, when I set the iteration number as n_iter = 30, the grid search method find the optimal model parameters.

parameters = { 'C' : [0.1, 1, 10, 100, 1000], 'gamma' : [0.0001, 0.001, 0.1, 10, 100] } # Some SVC parameters
ssscv = StratifiedShuffleSplit(y_train, n_iter=30, test_size=0.1) # 1. Let's build a stratified shuffle object

```
Final Model:
Best params: {'C': 1, 'gamma':
0.1}
Prediction time (secs): 0.0058
F1 score for training set:
0.971698113208
Prediction time (secs): 0.0017
F1 score for test set:
0.791946308725
```