Project Report
<Student Intervention>

Jihun Mun Brian

1. Classification V.S. Regression. Which one is more proper to apply in this project?

In this supervised machine learning project, I will use Classifier algorithm. Our data set has two parts, features part and result part. Our result set can be classified into 2 groups, passed or not. Also given features set with unknown results, we will predict result passed or not. So it is classification problem.

2. **Training and Evaluating Models**

Our data set is high dimensions. It has 30 features. And also the number of data set is not that big. So I choose 3 models as following.

- KNearestNeighbors Algorithms
- Decision Tree Algorithms
- SVM - SVC Algorithms

From now on, I will analyze the pros and cons about these 3 models.

1) KNearestNeighbors Algorithm

PROS	CONS
No training involved : We don't need to train our model. New example can be added anytime without time cost!!	Instead, querying time of the predicting data is expensive and slow. => O(#examples * # dimensions)
Very powerful, and complexity of parameter is simple. The only thing what we have to do is decide n_neighbors.	

To sum up, low train cost (O(1)) and expensive run time (O(#examples * #dimensions)). But the result would be not bad. And number of our example is not large, so I choose this model.

2) DecisionTree Algorithm

PROS	CONS
Querying time(Running time)is very fast.	Training time and complexity can be expensive
Normally Decision tree is very interpretable. Other algorithm is back box algorithm, but this is white box algorithm.	It is easy to overfitting. So you should prune when you make decision tree algorithm.

DecisionTree algorithm is very interpretable and once it is trained, the running time would be very fast. But you should be careful about managing complexity(overfitting problem).

3) SVM-SVC Algorithm

PROS	CONS
It is very effective when data set has high dimensional space. Also memory effective.	Training time is very long when the data set is large. So normally It is very ineffective with large number of training sets.
It works really well when there are a clear margin space between classification.	It is very weak at noise and overlapping.

SVM-SVC algorithm is very expensive algorithm(training time is very long). But in our case, the data set has high dimensionality and it contains small number of examples. So this algorithm is very suitable for our situation.

3. Choosing The Best Model

At this section I will choose the best model among the above 3 algorithms. To do this Let's compare the three tables below.

KNearestNeighbors Algorithm	Training Set Size		
	100	200	300
Training Time(sec)	0.001	0.002	0.001
Prediction Time(sec)	0.003	0.003	0.006
F1 score for training set	0.865853658537	0.836012861736	0.858447488584
F1 score for test set	0.773333333333	0.767123287671	0.785185185185

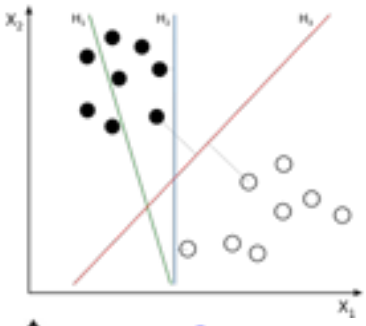
DecisionTree Algorithm	Training Set Size		
	100	200	300
Training Time(sec)	0.001	0.001	0.002
Prediction Time(sec)	0	0	0
F1 score for training set	0.979310344828	0.898360655738	0.883295194508
F1 score for test set	0.755905511811	0.780141843972	0.794326241135

SVM-SVC Algorithm	Training Set Size		
	100	200	300
Training Time(sec)	0.001	0.003	0.006
Prediction Time(sec)	0.001	0.002	0.004
F1 score for training set	0.88622754491	0.87898089172	0.864516129032
F1 score for test set	0.79746835443	0.81045751634	0.807947019868

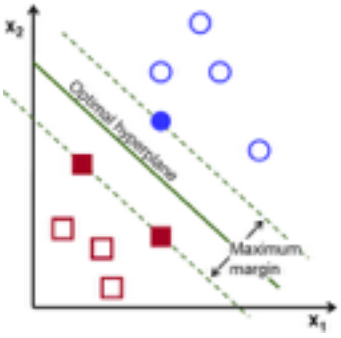
1) Their performance result are all similar. KNN and SVM-SVC algorithms' performance seems to be slightly better than decision tree algorithm. I strongly believe that Support Vector Classifier is the most suitable model for our data set. It shows the best F1 score. And also as I told you, It is very fit on small number of data set with high dimensionality. Also based on the available data (small number!!), and training and prediction time cost is not expensive. So Support Vector Classifier is the best model!!

2) Now let's talk about SVM-SVC algorithm more detain in layman's term. Basically this algorithm classify the data set. Given the training data set which target is already classified, this algorithm involves finding the hyperplane (hyperplane can be line or 2D dimensional plane, …, more generally, hyperplane is n-1 dimensional space when the data set contains n features). That hyperplane should be clearly divide the data set to classified groups with maximum margin. We call the hyperplane with maximum margin is a optimal hyperplane of model. So finding the optimal hyperplane of model is the purpose of SVC algorithm. I will explain in more detail about hyperplane and maximum margin.

* (Optimal) hyperplane and maximum margin

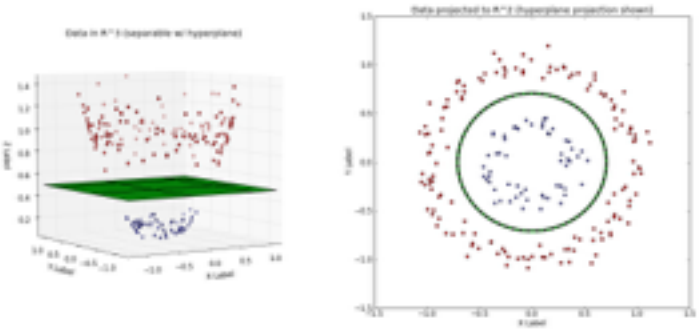


As you can see on left picture, H2 and H3 can be hyperplane. Both are exactly divide the data set which is already classified. But the line 'H3" is the optimal hyperplane. Because margin of H3 is bigger than of H2. So H3 has the maximum margin. So If we apply SVC algorithm in this case, we can obtain optimal hyperplane like H3.



Maximum margin can be sounds little vague. If so, you can check the left picture.
Maximum margin is the minimum orthogonal (to hyperplane) distance between support vector which belong to different classified groups.

Above cases are 2-dimensional features case. So the hyperplane has less then 2 dimensional space. Following examples show the high dimensional hyperplane example.



From now on I will talk about more on training process. Let's start with 2D with linear hyperplane case (because at first this example is easy to understand). So this algorithm will search the line like 'ax + by = c' which is separating line. Among the hyperplanes it will find optimal hyperplane as I mentioned at second picture. This is a process of training, finding optimal hyperplane by calculating the orthogonal distance between support vectors.
Sometimes we need curved hyperplane, not linear. In these cases, we use kernel function. When we see the fourth picture, the optimal hyperplane is circle, $X^2 + y^2 = R^2$. Like wise we can add some polynomial terms like X^2, X^3, … instead of linear terms. For more complex examples, we can apply gaussian kernel terms or many other trick kernels.

Once we trained our model, It is very easy (cheap) to predict new data. At training process, we obtained the optimal hyperplane. Given new data set, we substitute our features to the equation of hyperplane and calculate which group contain the new features.