MLND P2 submission

Q4. Training and Evaluating Models

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

- What is the theoretical O(n) time & space complexity in terms of input size?
- What are the general applications of this model? What are its strengths and weaknesses?
- Given what you know about the data so far, why did you choose this model to apply?
- Fit this model to the training data, try to predict labels (for both training and test sets), and measure the F_1 score. Repeat this process with different training set sizes (100, 200, 300), keeping test set constant.

Produce a table showing training time, prediction time, F_1 score on training set and F_1 score on test set, for each training set size.

Note: You need to produce 3 such tables - one for each model.

Sample	Training set size		
	100	200	300
Training time (secs)			
Prediction time (secs)			
F1 score for training set			
F1 score for test set			

ANS Below

1. SVC - Support Vector Classification

- What is the theoretical O(n) time & space complexity in terms of input size?
    - Time complexity: $O(n^4)$
    - Space complexity: O(n)
    - Reference: http://scikit-learn.org/stable/modules/svm.html#complexity
- What are the general applications of this model? What are its strengths and weaknesses?
    - SVM's work great when the data is separable using a hyperplane.
    - Strengths:
        - SVMs generate a classifier that is O(n) in space complexity and hence is portable and fast (can run on embedded systems)
    - Weekness:
        - It does not work well when the data is not linearly separable.
        - When there are too many features, the data has to be ported into a higher dimensional space. Which can grow to become quite large.
        - When the data has to be updated and re-trained often, SVMs can be slower than some other options.
- Given what you know about the data so far, why did you choose this model to apply?
    - We can see that a lot of features in data are linearly separable (travel-time, study-time). Therefore, this data is easily separable by a hyperplane.
- Fit this model to the training data, try to predict labels (for both training and test sets), and measure the F_1 score. Repeat this process with different training set sizes (100, 200, 300), keeping test set constant.

SVC	Training set size		
	100	200	300
Training time (secs)	0.001	0.003	0.007
Prediction time (secs)	0.001	0.002	0.004
F1 score for training set	0.877697841727	0.867924528302	0.876068376068
F1 score for test set	0.774647887324	0.781456953642	0.783783783784

2. Decision Trees

- What is the theoretical O(n) time & space complexity in terms of input size?
    - Time Complexity: $O(n^2 \log_n)$
    - Space Complexity: O(n^2)
    - Reference: http://scikit-learn.org/stable/modules/tree.html#complexity
- What are the general applications of this model? What are its strengths and weaknesses?
    - This model is used to split the data into multiple subclasses.
    - Strengths:
        - Great for binary classification
        - Has excellent classification for data already seen before (training set)
    - Weaknesses:
        - Can get really complicated , and will tend to overfit given the many features.
        - Has a high space complexity, thus is not very portable.
- Given what you know about the data so far, why did you choose this model to apply?
    - The data has a lot of binary features. So Decision tree would fit and create subclasses efficiently.
- Fit this model to the training data, try to predict labels (for both training and test sets), and measure the F_1 score. Repeat this process with different training set sizes (100, 200, 300), keeping test set constant.

	Training set size		
	100	200	300
Training time (secs)	0.001	0.001	0.002
Prediction time (secs)	0.000	0.000	0.000
F1 score for training set	1	1	1
F1 score for test set	0.628099173554	0.753846153846	0.633333333333

3. Gaussian Naive Bayes

- What is the theoretical O(n) time & space complexity in terms of input size?
    - Time Complexity: $O(n^2)$
    - Space Complesity: $O(n^2)$

- What are the general applications of this model? What are its strengths and weaknesses?
    - Strengths:
        - Simple probabilistic model works well for binary features.
    - Weaknesses:
        - It assumes that the values are independent
- Given what you know about the data so far, why did you choose this model to apply?
    - The data encountered in the future will be similar to the data being trained on. So NB can work well with data it has seen before.
- Fit this model to the training data, try to predict labels (for both training and test sets), and measure the F_1 score. Repeat this process with different training set sizes (100, 200, 300), keeping test set constant.

	Training set size		
	100	200	300
Training time (secs)	0.001	0.001	0.001
Prediction time (secs)	0	0	0
F1 score for training set	0.846715328467	0.840579710145	0.80378250591
F1 score for test set	0.802919708029	0.724409448819	0.763358778626

Q5. Choosing the Best Model

- Based on the experiments you performed earlier, in 1-2 paragraphs explain to the board of supervisors what single model you chose as the best model. Which model is generally the most appropriate based on the available data, limited resources, cost, and performance?
    - I would say that SVC is the best model. SVC have low space complexity and can efficiently classify real time. Since we have just two classes to be classified, there is just a single hyperplane that needs to be determined.

- In 1-2 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 make a prediction).
    - Support Vector Machine uses the data and tries to find a function, that can split data into classes.

      It will attempt to find a hyperplane that divides the data, It does so by finding the closest two points. It determines a vector connecting them, and then the plane that bisects this vector.
- 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 F_1 score?
    - 0.783783783784