Homework Assignment #2 - Machine Learning Lab (Working with Data)
EE382V Activity Sensing and Recognition - Fall 2016

*Due **Sept 20th 2016 @ 5PM***

Answer the 3 questions below (and associated sub-questions). The datasets q1_data.csv, q2_data.csv, and q3_data.csv can be found on Canvas (Files > Homework > ML Lab).

Turn-in your homework as a zip file including your answers, plots and code. Label which files correspond to which answers very carefully. Make it intelligible. If this is not clear, you will not receive credit for your solution.

Q1. *(40 pts)* The dataset in q1_data.csv is comprised of 1000 instances, each with 2 features (F1 and F2) and a binary label (0 or 1). The two features are related in a particularly interesting way.

a) Find out what this relationship looks like by plotting a 2-D graph of the two features. You can use Python's matplotlib for this task or any other plotting tool of your choice. Include the plot in your answer. *(5 pts)*

b) Using scikit-learn, fit a logistic regression model to the dataset and evaluate its performance (accuracy) with 10-fold cross-validation. You should report performance using the accuracy measure, averaged across all cross-validation runs. Include your source code. *(10 pts)*

c) Can you think of a way to improve the performance of the model while still employing the logistic regression algorithm? If so, describe how, include your code and present performance results. (Hint: think about creating new features based on F1 and F2). *(15 pts)*

d) Fit a Random Forest model to the dataset and based on what you find out, discuss why you think it performs better or worse than logistic regression. Include your source code. *(10 pts)*

Q2. *(40 pts)* Support Vector Machines (SVM) are classifiers that aim to maximize the margin separating classes. In addition to linear classification, SVMs can also be used in non-linear classification by using kernel functions. In this case, instances are mapped to a high-dimensional space and separated using one or more hyperplanes.

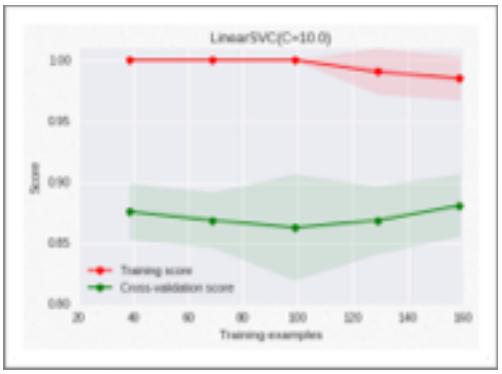
a) Plot the dataset q2_data.csv and indicate whether you think the two dataset classes are linearly separable. Include the plot in your answer. *(5 pts)*

b) Using scikit-learn, fit an SVM model with a **linear** kernel to the dataset and evaluate its performance (accuracy) with 10-fold cross-validation. You should report performance using the accuracy measure, averaged across all cross-validation runs. Include your source code. *(10 pts)*

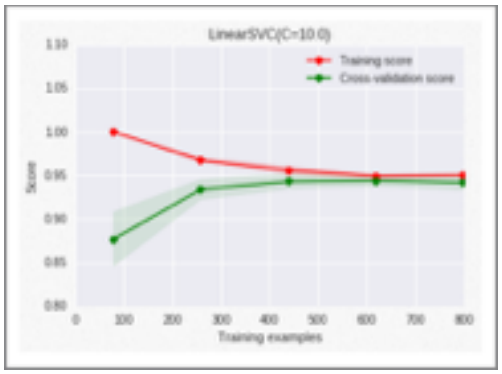
c) Now, fit an SVM model with a **non-linear** kernel to the dataset and evaluate its performance (accuracy) with 10-fold cross-validation. You should report performance using the accuracy measure, averaged across all cross-validation runs. Include your source code. *(10 pts)*

d) Fit a Random Forest model to the q2_data.csv dataset and discuss why you think it performs better or worse than SVM (with linear and non-linear kernel), and how the number of trees in the forest affect the performance results. Include your source code. *(15 pts)*

Q3. *(20 pts)* While evaluating a classifier, it is important to understand whether it is overfitting on the training data. One way to check is by computing a classifier's learning curve; it shows a measure of performance (e.g., accuracy) as the classifier is trained and tested (cross-validated) with increasingly more data. If the curves do not converge, there is overfitting. For example:



Overfitting



Not Overfitting

You can plot learning curve results for a Linear SVM classifier on the q3_data.csv using the following code based on scikit-learn:

```
Xy = np.loadtxt("q3_data.csv", delimiter=",")
X = Xy[:,:20]
y = Xy[:,20]

train_sizes_abs, train_scores, test_scores = learning_curve(LinearSVC(C=10),
X, y, cv=5, n_jobs=1, train_sizes = np.linspace(.05, 0.2, 5))

print(np.mean(train_scores, axis=1))
print(test_scores_mean = np.mean(test_scores, axis=1))
```

Provide source code showing one way to reduce overfitting in this example without modifying the classification algorithm (i.e., LinearSVC). (Hint: Think about training data size, regularization and SVM parameterization). *(20 pts)*