Problem 1

	Decision or Regression Tree	**Linear Regression (Normal Equations)**	**Linear Ridge Regression (Normal Equations)**	**Linear Regression(Gradient Descent)**	**LogisticRegression(Gradient Descent)**
Spambase	Train ACC: 92% Test ACC: 87.5%	Train ACC: 89.2% Test ACC: 88.6%	Train ACC: 90.3% Test ACC: 89.17%	Train ACC: 90.4% Test ACC: 89.67%	Train ACC: 90.72% Test ACC: 90.01%
Housing	Train MSE: 14.1 Test MSE: 39.6	Train MSE: 22.08 Test MSE: 22.6	Train MSE: 22.08 Test MSE: 22.44	Train MSE: 22.09 Test MSE: 22.4	N/A . - not classification task

C.
Linear Regression (Average of 3 Folds)

	Pred No	Pred Yes
Actual No	855.33	70.66
Actual Yes	68.66	521.33

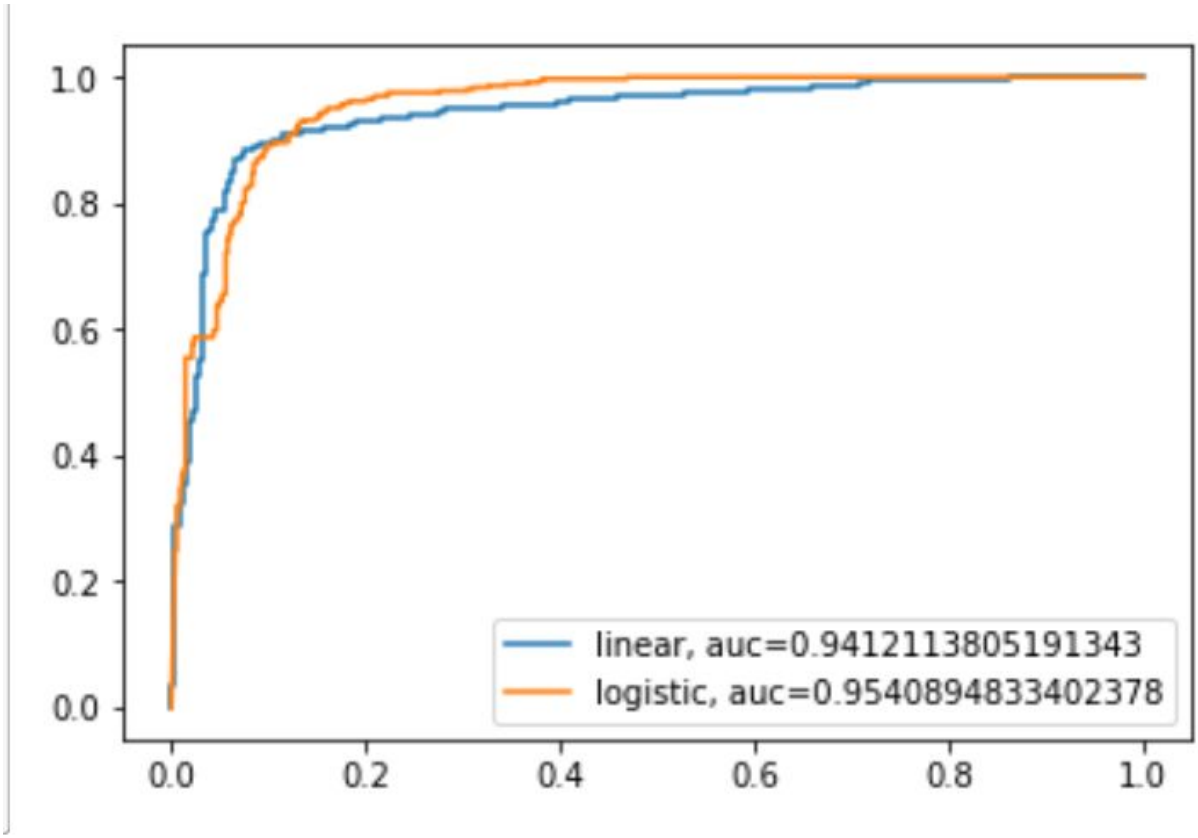
Logistic Regression (Average of 3 Folds)

	Pred No	Pred Yes
Actual No	832.66	93.33
Actual Yes	56	548

Decision Tree (Average of 3 Folds)

	Pred No	Pred Yes
Actual No	829.33	89
Actual Yes	61.66	550

D.



Problem 3

Andrew Ng Summary

A lot of times in life, people try to solve problems before having a clear understanding of the issues. Given that the whole point of machine learning is using data to solve problems by having computers “learn” that understanding, you would think that machine learning engineers

would use that sort of approach to debugging their own algorithms. According to Andrew Ng, that is not the case.

In his lecture, Andrew Ng lists a plethora of “traditional” methods for debugging machine learning algorithms, such as playing with the features, using more iterations, playing with parameters etc. He states that while one can solve their issues this way, but it would be extremely time intensive and in the end the outcome is more about luck than anything. What Andrew Ng suggests is instead of jumping into solutions, first understand the problem more and then come up with more specific solutions.

Two common diagnostics he shows in the lecture are variance and bias. High variance typically indicates the algorithm is overfitting, while high bias shows you have too few features. You may have high variance if your training error is much lower than your test error, and high bias usually happens if both training and test errors are high. He also gives other examples of diagnostics to check, and how to fix the issues that are revealed. However at the end of the day, he says that the most useful diagnostics are the ones that you come up with for your problem.

Error analysis Andrew Ng says is a good practice, as understanding the source of your errors can help one reduce them. Most algorithms are comprised of many different components, Andrew Ng suggests decomposing the “pipeline” and trying to evaluate how much each component really brings to the table via a ablative analysis.

Last but not least, Andrew Ng gives two different approaches to solving problems. One is more along the lines of traditional waterfall software development method, and the other is closer to the now extremely popular agile software method. Both have their benefits, but like most things in life you’ll have to weigh the pros and cons for yourself to see what’s best.

A Few Useful Things to Know about Machine Learning

- Machine learning is an attractive solution for problems, there are a lot of algorithms to choose from
- Algorithms are typically composed of three parts
- Representation- There are different categories of algorithms (see table)
- Evaluation- How you score the algorithms
- Optimization- Need a way to find the best parameters for the algorithms
- Major key to machine learning is that models need to be able to generalize beyond the scope of the training data
- Cross validation can help with generalization especially if there isn't a lot of data
- Because the objective is just an approximation, we don't need to fully optimize, sometimes the local minima will be better than the global
- Sometimes data is not enough to generalize, have to use SMEs sometimes
- Overfitting is broad, dive into variance and bias to get at the real issues
- Different classifiers can tend to run into the same respective problems
- Cross validation and regularization can be useful to fight overfitting
- Noise is not required to have high variance
- In high dimensional spaces, many algorithms break, "Curse of Dimensionality"
- A lot of things seem good in clean academic settings, but break in reality
- Feature engineering is incredibly important, often more time consuming than actual machine learning
- Machine learning projects are often highly iterative
- A lot of the times a model with more data will outperform a "better" model with less data to train on

- Model ensembles are frequently used and effective, BMA not worth your time though
- Simplicity is good but isn't everything (model ensembles!)
- Some representations cannot be learned (yet!)
- Correlation != causation and all the implications of that