GENERAL ASSEMBLY

DATA SCIENCE
11 WEEK PART TIME COURSE

Week 7 – Ensembling with Decision Trees
Wednesday 4th May 2016

AGENDA

1. Guest Speaker - Sidney Minassian (Contexti)
2. Review of Decision Trees
3. Ensembling
4. Bagging
5. Random Forest
6. Boosting
7. Lab
8. Other Uses
9. Discussion

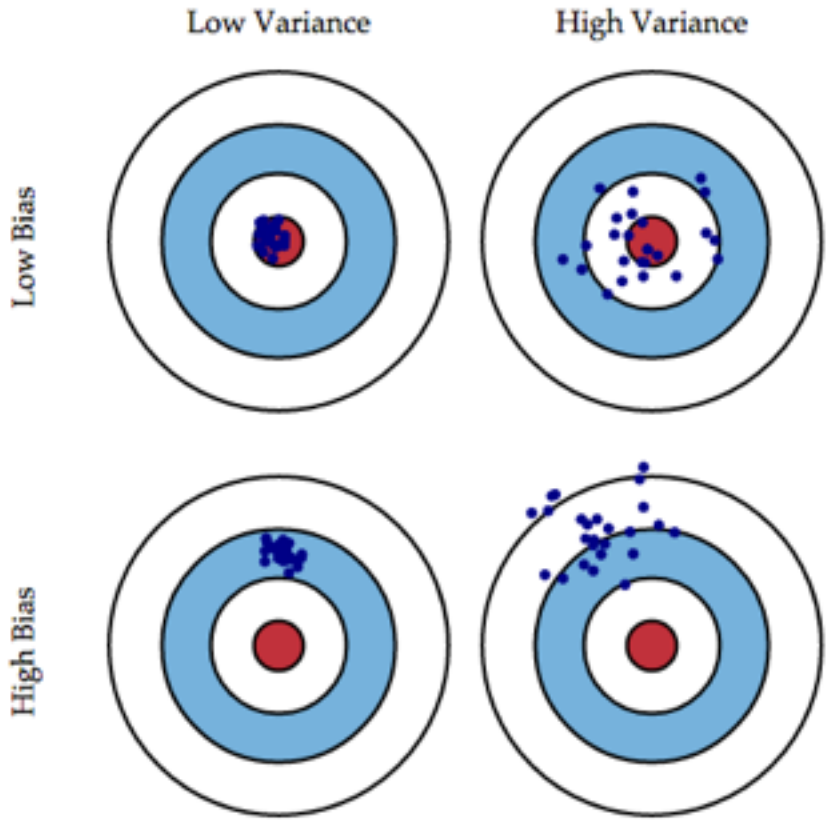
DATA SCIENCE PART TIME COURSE

DECISION TREES

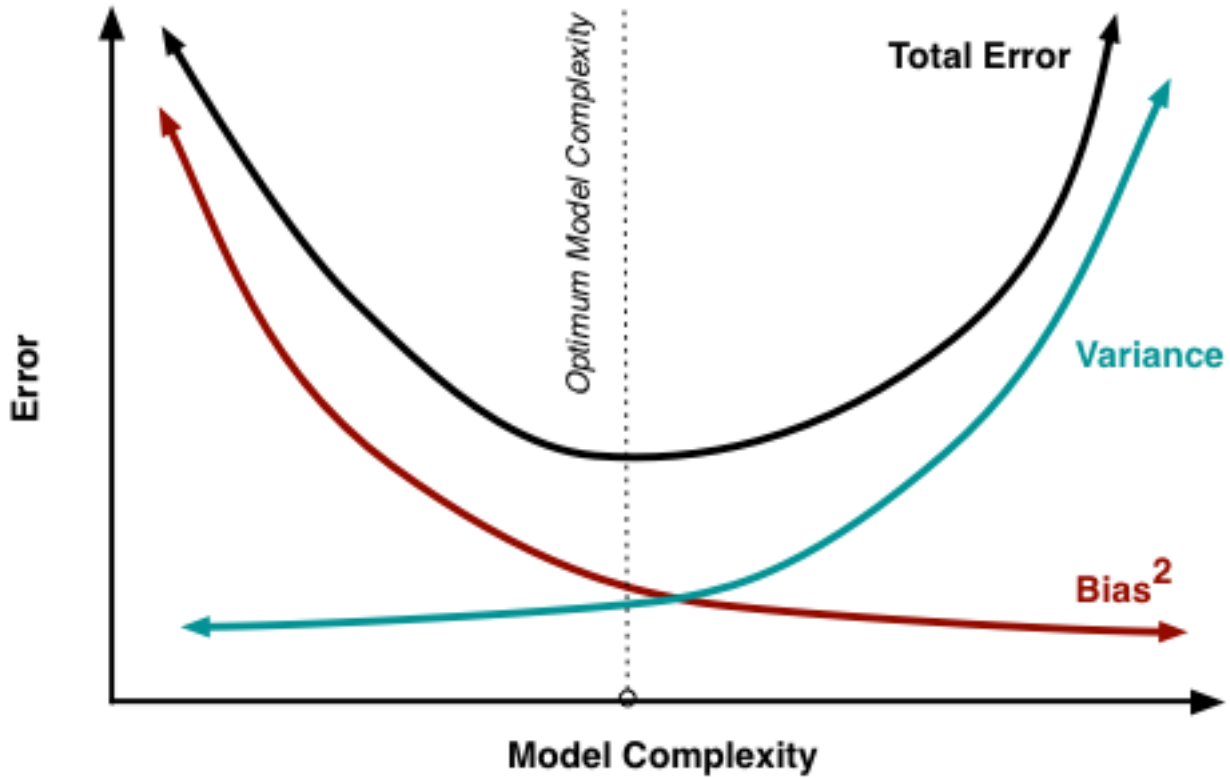
- A supervised learning technique that can be used for classification or regression.
- Visually engaging and easy to interpret.
- Foundation for getting into very powerful techniques.
- Great for explaining to people!

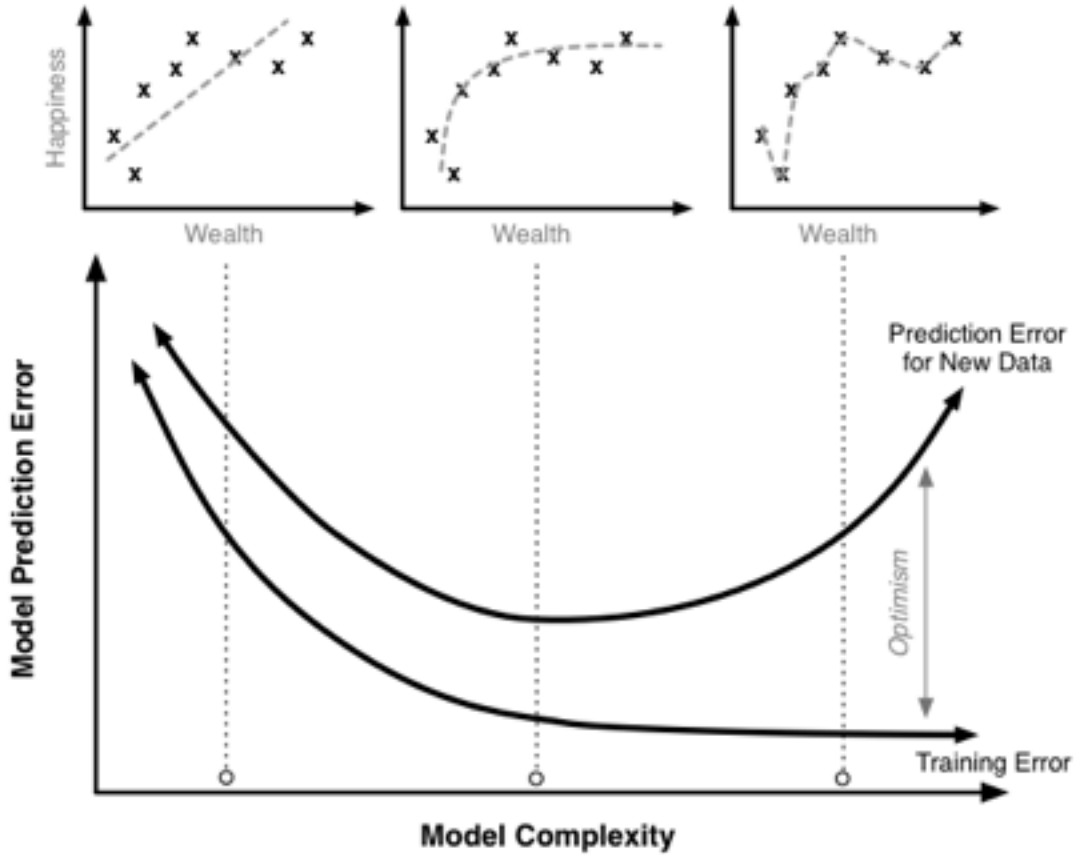
- Scans for a feature to split on that results in the greatest separation between classes in the resulting nodes.
- Non-linear
- Greedy process
- Splits within splits
- For a classification tree, we predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs.
- We naturally get combinations of features used for our prediction.

- Prone to overfitting.
- Predictive power is lower in comparison to many other modern techniques.
- They suffer from high variance


Low Variance
High Variance
Low Bias
High Bias

BIAS – VARIANCE TRADEOFF




Happiness
Wealth
Wealth
Wealth
Model Prediction Error
Prediction Error
for New Data
Optimism
Training Error
Model Complexity

HOW DO WE IMPROVE DECISION TREES ?

DATA SCIENCE PART TIME COURSE

ENSEMBLING

Ensemble learning (or "ensembling") is simply the process of combining several models to solve a prediction problem, with the goal of producing a combined model that is more accurate than any individual model.

For classification problems, the combination is often done by majority vote.

For regression problems, the combination is often done by taking an average of the predictions.

For ensembling to work well, the individual models must meet two conditions:

- Models should be accurate (they must outperform random guessing)
- Models should be independent (their predictions are not correlated with one another)

DATA SCIENCE PART TIME COURSE

BAGGING

- Bootstrapped Aggregation = Bagging
- The bootstrap is a statistical technique that is used to quantify the uncertainty of a model
- Bootstrap samples are simply random samples with replacement

What is the bagging process?

1. Take repeated bootstrap samples (random samples with replacement) from the training data set
2. Train our method on each bootstrapped training set and make predictions
3. Average the predictions

This increases predictive accuracy by reducing the variance, similar to how cross-validation reduces the variance associated with the test set approach (for estimating out-of-sample error) by splitting many times an averaging the results.

Specifically to Decision Trees

1. Grow B regression trees using B bootstrapped training sets
2. Grow each tree deep enough that each one has low bias
3. Every tree makes a numeric prediction, and the predictions are averaged (to reduce the variance)

What value should be used for B?

HOW DO WE IMPROVE BAGGED DECISION TREES ?

DATA SCIENCE PART TIME COURSE

RANDOM FORESTS

Random Forests is a slight variation of bagged trees that has even better performance! Here's how it works:

- Exactly like bagging, we create an ensemble of decision trees using bootstrapped samples of the training set.
- However, when building each tree, each time a split is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors. The split is only allowed to use one of those m predictors.

However, when building each tree, each time a split is considered, a

random sample of m predictors

is chosen as split candidates from the

full set of p predictors.

The split is only allowed to use one of those

m predictors.

Notes:

- A new random sample of predictors is chosen for every single tree at every single split.
- For classification, m is typically chosen to be the square root of p. For regression, m is typically chosen to be somewhere between p/3 and p.

What's the point?

- Suppose there is one very strong predictor in the data set. When using bagged trees, most of the trees will use that predictor as the top split, resulting in an ensemble of similar trees that are "highly correlated".
- Averaging highly correlated quantities does not significantly reduce variance (which is the entire goal of bagging).
- By randomly leaving out candidate predictors from each split, Random Forests "decorrelates" the trees, such that the averaging process can reduce the variance of the resulting model.

Although bagging increases predictive accuracy, it decreases model interpretability because it's no longer possible to visualize the tree to understand the importance of each variable.

- To compute variable importance for bagged regression trees, we can calculate the total amount that the mean squared error is decreased due to splits over a given predictor, averaged over all trees.
- A similar process is used for bagged classification trees, except we use the Gini index instead of the mean squared error.

Bagged models have a very nice property: out-of-sample error can be estimated without using the test set approach or cross-validation. How it works:

- On average, each bagged tree uses about two-thirds of the observations. For each tree, the remaining observations are called "out-of-bag" observations.
- For the first observation in the training data, predict its response using only the trees in which that observation was out-of-bag. Average those predictions (for regression) or take a majority vote (for classification).
- Repeat this process for every observation in the training data.
- Compare all predictions to the actual responses in order to compute a mean squared error or classification error. This is known as the out-of-bag error.

WHAT ELSE COULD WE DO TO IMPROVE DECISION TREES ?

DATA SCIENCE PART TIME COURSE

BOOSTING

Boosting involves the base learners being built sequentially (using previous models). The aim is to reduce the bias of the combined estimator.

Compare this to something like bagging where the models are built independently of one and other.

AdaBoost works by building sequential models on re-weighted versions of the data. Initially all data points have the same weighting = 1/N.

After the first model the error on the points is evaluated and the weighting for each observation is adjusted. So if we fit a model and an observation has a large difference between the observed and predicted we give that observation a higher weighting and run the model again.

This means observations that are difficult to predict become more influential in the model.

Gradient Boosting is a technique that are sequentially fitted that minimise a loss function. For regression this might be the square loss. So we would be updating the model based on Gradient Descent.

This differs to AdaBoost where the boosting was performed by re-weighting the original observations.

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting(also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment(Hadoop, SGE, MPI) and can solve problems beyond billions of examples.

- https://github.com/dmlc/xgboost

DATA SCIENCE PART TIME COURSE

git remote -v

git remote add upstream https://github.com/ihansel/SYD_DAT_3.git

git remote -v

git fetch upstream

git checkout master

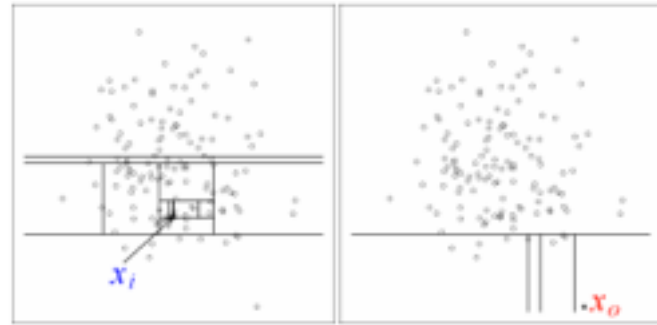
git merge upstream/master

OR git reset –hard upstream/master

OTHER USES

- The idea is to create Synthetic data (fake data that is based on the original data set)
- Then combine this fake data with your original data set
- Assign a ‘Synthetic_data’ with 1 for the synthetic data and 0 for real data
- Run the Random Forest model trying to predict ‘Synthetic_data’
- You will have a ‘distance metric’ or ‘proximity’ which is basically how many times the samples occur in the same terminal nodes.

ISOLATION FOREST FOR OUTLIER DETECTION

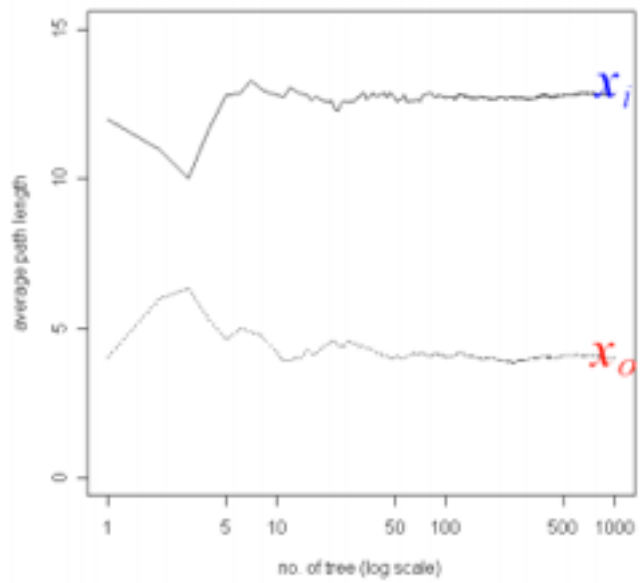


(a) Isolating x_i (b) Isolating x_o

Isolation Forest

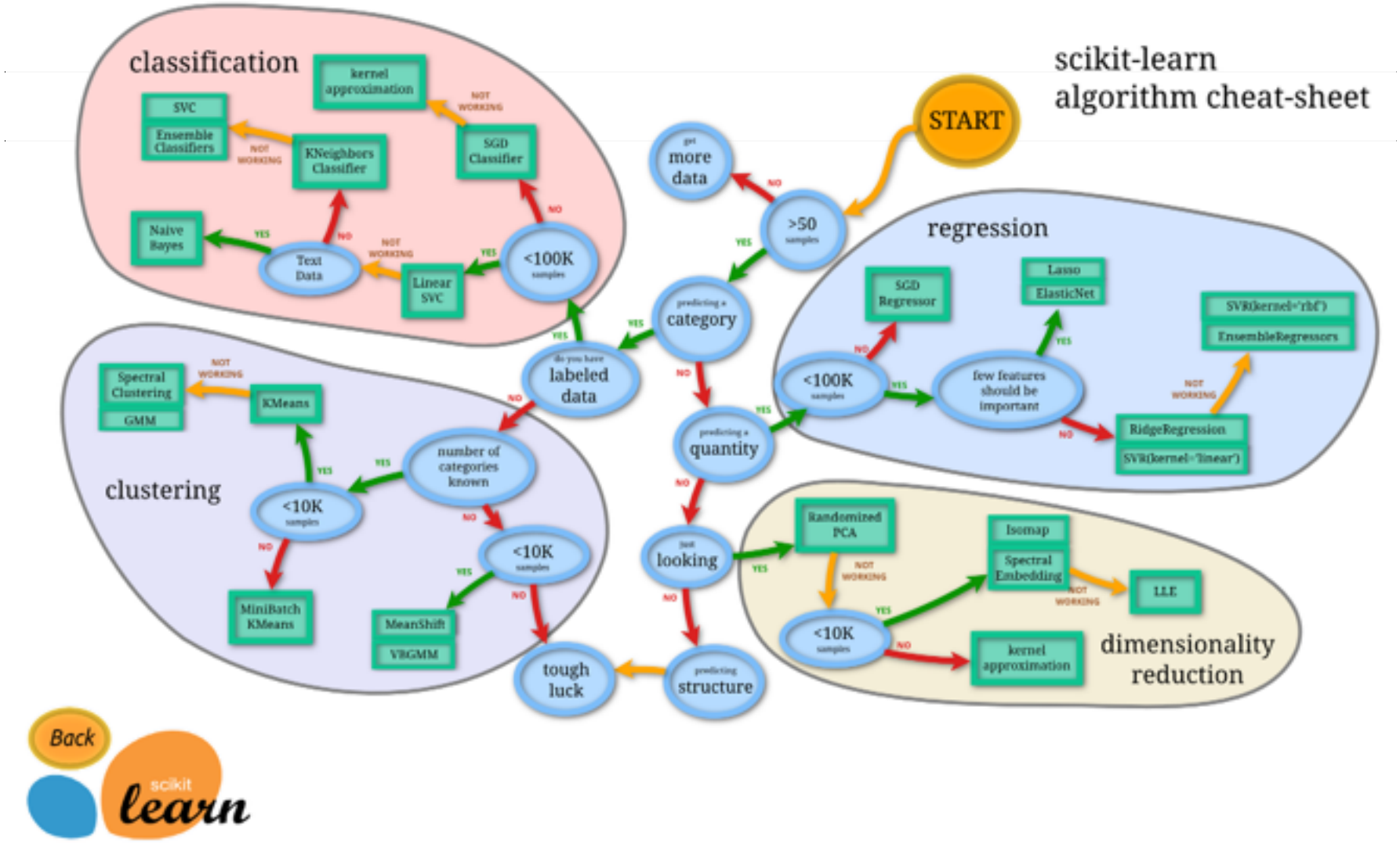
Fei Tony Liu, Kai Ming Ting
Gippsland School of Information Technology
Monash University, Victoria, Australia
{tony.liu},{kaiming.ting}@infotech.monash.edu.au

Zhi-Hua Zhou
National Key Laboratory
for Novel Software Technology
Nanjing University, Nanjing 210093, China
zhouzh@lamda.nju.edu.cn



(c) Average path lengths converge

- Extract all the rules from all the trees and use them as features in an elastic net model.
- Nice way to build a simpler more interpretable model


scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
NO
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
<100K samples
NO
YES
few features should be important
YES
NOT WORKING
RidgeRegression
NO
SVR(kernel='linear')
YES
predicting a category
YES
do you have labeled data
NO
predicting a quantity
YES
NO
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
YES
<10K samples
NO
NO
<10K samples
YES
MiniBatch KMeans
MeanShift
VBGMM
NO
just looking
YES
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
NO
predicting structure
tough luck
Back
scikit learn

DISCUSSION TIME

- Hackday
- Rest of the course
- Tasklist

DATA SCIENCE – Week 7 Day 2

HACKDAY

- Saturday 7th May 10:00 – 3:00
- Review topics completed done so far
- AWS
- Projects

COURSE REMAINDER

- Practical Skills (Cloud)
- Advanced Topics (Time Series, Graph Analysis, Neural Networks, Natural Language Processing)
- Course Review
- Project Presentations (Final 2 sessions)
- Lots of guest presenters

Task List

- [ ] Email me anything you'd like to review (previously been covered) by Thursday for Saturday
- [ ] Email me anything you'd like to cover (not covered) by Thursday for Saturday
- [ ] Work on your project