Final Project

Matthew Newell, Michael Quintana, and Albert Simorangkir

Introduction

For this project, we first aimed to find the best model to predict election results using logistic regression, linear discriminant analysis, and quadratic discriminant analysis. From these three methods, we found that using the variables in the New_data set within a linear regression model (with an optimal threshold set using an ROC curve) produced the most accurate results, with approximately a 92% predictive accuracy of predicting a Trump win and an 90% predictive accuracy in predicting a Clinton win. We then created a decision tree and compared its prediction performance with the previously mentioned methods by analyzing its misclassification rates.

The second task we pursued was to model the probability of a win by one candidate in different clusters of counties, to see if clustering before making predictions results in superior predictions. We created our clusters of counties using the K-means clustering method. This would be known as unsupervised learning because we are trying to find a grouping in our data with only the features. After we created the clusters we performed supervised learning on K=4, K=2, and K=1 (a.k.a. the base case). We used a Boosting model, and a Logistic Regression Model. From these supervised learning methods, we found that K=4, K=2, K=1, all predicted Trump to win every single cluster of counties in both the boosting and logistic regression method. We found that clustering before supervised learning did give us superior results with our Logistic Regression model. While our results were inferior with clustered data when looking at our Boosted model results.

Methods

Method 1 To accurately predict the winning candidate in each county, we created logistic regression, linear discriminant analysis, and quadratic discriminant analysis models and compared the misclassification rates. We used the census information as covariates for predicting the probabilities for whether Trump or Clinton would win a county (response). LDA and QDA require two key assumptions which are that the covariates are approximately multivariate Gaussian and that the observations are independent and identically distributed. Whereas LDA and QDA make assumptions about the distribution of predictor variables, logistic regression does not. The difference between LDA and QDA is that LDA assumes the covariances are equal across groups.

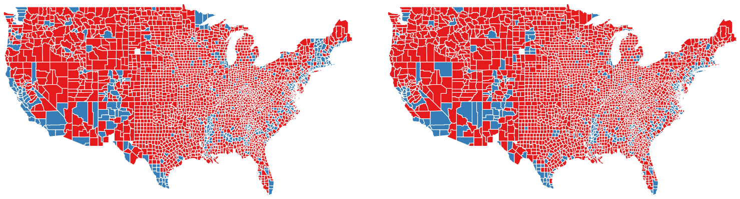
Method 2 We preform K-means clustering to group the county observation by their features. Clusters are formed so that the features are more similar within clusters and least similar between cluster. For K-means clustering, the similarity measure is based on distance. K-means clustering finds K center points that minimize the average distance between points within a cluster. After we performed a supervised learning ensemble method, known as a boosted model on the K cluster of counties. In this model we train a tree estimator on training data and obtain weights. With those weights we get error measurements and continue iterating to get our results. We also performed the previously mention logistic regression on our clustered data.

K-means finds a local solution to:

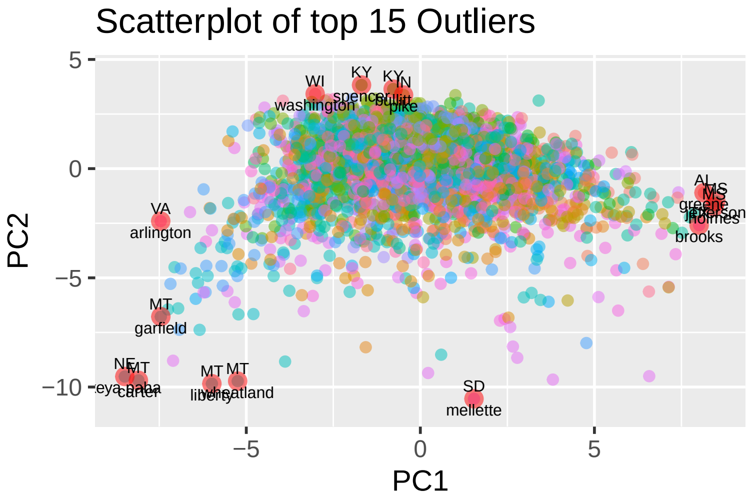
$$\mathbf{c}_k = \arg\min \left\{ \sum_{i=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2^2 \right\}$$

Credit: Trevor Ruiz (Week-7-Clustering) for Latex

Visualizations For Method 1 & 2



The left US map is the actual winner per county in the 2016 election. The map on the right is the predicted results of the 2016 method using the a Logistic Regression Model in Method 1. As you can see there are only very small changes between the predicted results and actual results. However this graph is a tad misleading and in fact our predictions are overfitted for this US map figure. This is because in order to produce the map, I needed a prediction for each county. So I had to make predictions on both the training and test data.

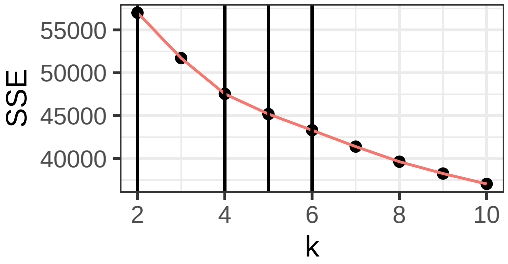


From our graph the only distinct outlier seems to be mellete, South Dakota. However when looking closely we notice that the its PC2 value isn't too extreme. While we don't have a particular issue removing this observation we also don't feel its completely necessary.

Table 1: 3 Most Extreme PC2 Values

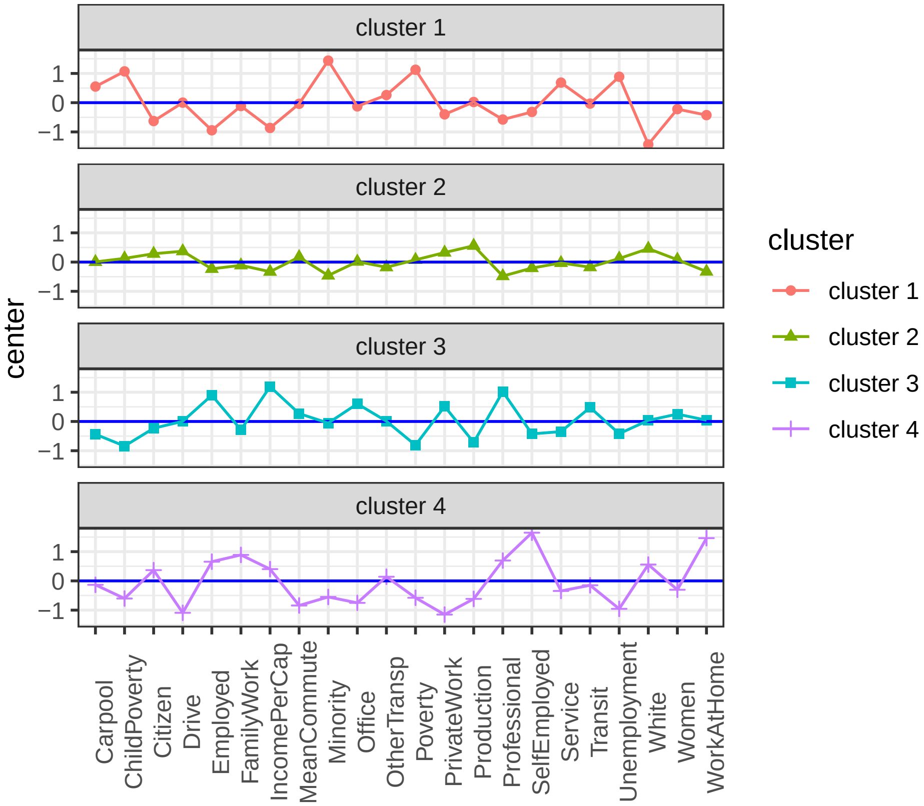
PC1	PC2	county	state
1.541	-10.54	mellette	SD
-5.988	-9.849	liberty	MT
-5.247	-9.73	wheatland	MT

Choosing K



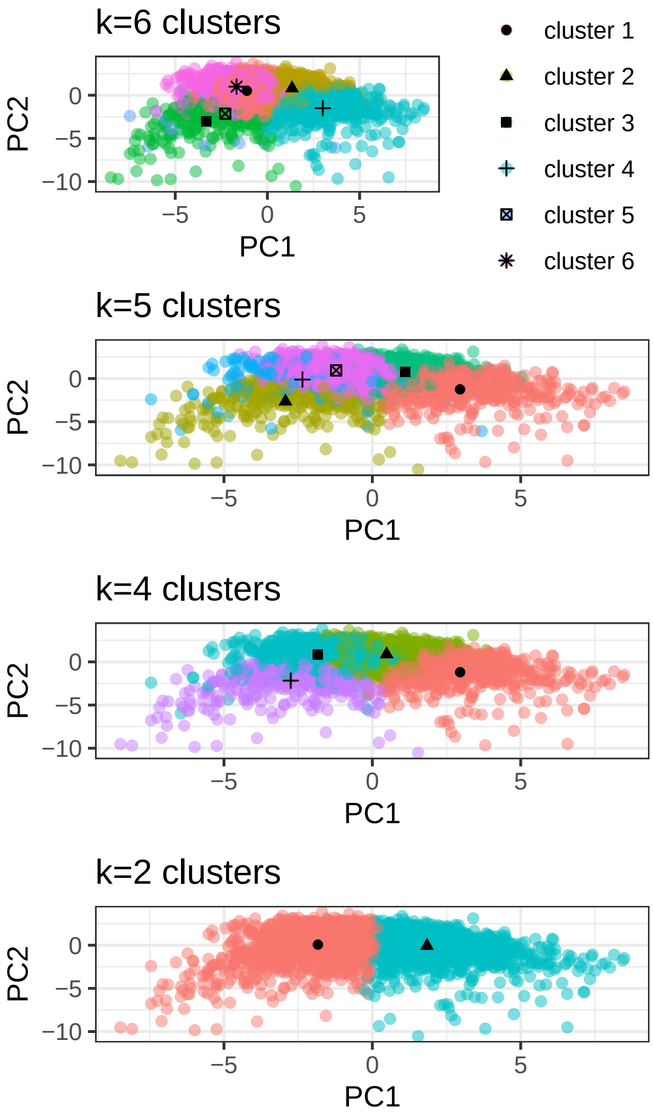
I want to try multiple K values for the clustering of counties, Nbclust function via 'Nbclust' library recommends K=2. However I wanted a slightly larger amount of clusters in order to have a more deatiled grouping of counties. There I'm also going to look at K=4,5,6

Centroids for K=4

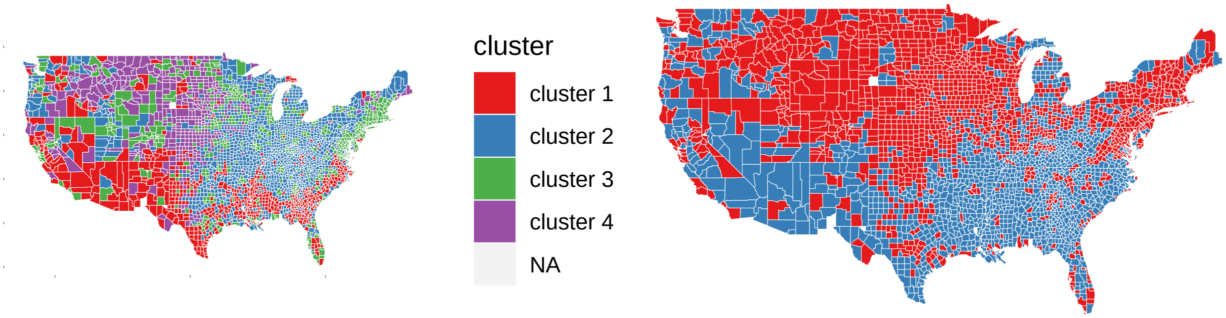


Centroid information for K=4, this is just so we can get an idea of the significant variables in the clustering.

~For Cluster 1, the most significant variables seemed to be ‘Minority’, ‘Poverty’, and ‘White’.~For Cluster 2, we see that no variable seemed to have extreme significance. If I had to pick I would say ‘White’, ‘Minority’, and ‘Production’.~For Cluster 3, ‘Professional’,‘Poverty’, ‘IncomePerCap’,’ and ‘Child Poverty’.~And finally for Cluster 4, ‘SelfEmployed’, ‘WorkAtHome’, ‘Drive’ and ‘Private Work’

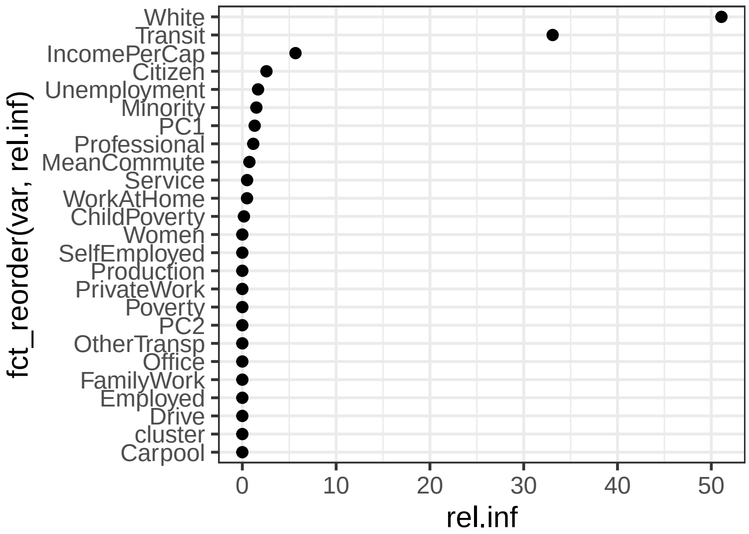


Here we see scatterplots of Clusters K=2, K=4, K=5, K=6 using PC1 & PC2. K=2 has a pretty well defined split, but like we previously mentioned we want more than 2 clusters. K=4 involves some mixture between the clusters, but for the most part there is a defined split between clusters. While K=5 & K=6 has too much mixture inbetween clusters in order for us to feel confident about there use in supervised learning. Therefore we will continue with K=4, K=2, and K=1 (a.k.a. the base case)



Here we see two US Maps with Clusters K=2 and Clusters K=4. This is to give us an idea of how the counties are clustered around the USA. For K=4 we see the clusters are fairly spread out (Besides a large group 1 cluster across the Southern USA.) K=2 is somewhat split into two cluster of the southern half of the United States and the northern half.

Influential Variables in Method 2 (Supervised Prediction K=1) I would have loved to show graphs of importance measures (variable influence) for K=4, but unfortunately we don't have the space for that many figures. Instead I did it for the base case K=1, to give us a general idea of the most influential variables in the model overall.



This is a plot of the variable influence measures for our boosting method. From this plot we see that the most significant variables driving our boosting predictions were 'IncomePerCap','Transit', and by far the most significant 'White'.

Employed	Citizen	Service	Professional	Production
267564080	2210716	1.504	1.335	1.259

From the coefficients of our Logistic Regression model, we see that the variables that our found to be the most significant in the total model are 'Employed', 'Citizen', 'Service','Professional', and 'Production'. Therefore we see our two models used in this method don't argee when it comes to the most influential variables in the model.

Results

Method 1

To predict the probability that either Trump or Clinton would win a county, we created a linear regression model, a LDA model, and a QDA model using all of the variables in the New_data data set. While all models have high predictive accuracy for Trump victories (around 94%-97%), the predictive accuracy for Clinton wins is between 68% and 71%, which is relatively low. In addition, the Trump misclassification rates are relatively high (around 30%). Therefore, we added an optimal threshold using ROC curves and produced new models for each method. Out of these three updated models, we found that the linear regression model with an optimal threshold produced the highest predictive accuracy with approximately 86% predictive accuracy with Trump wins and 93% predictive accuracy with Clinton wins. Including the optimal threshold also significantly reduced the misclassification rate of Trump wins. From our model's high predictive accuracy, we can conclude that the covariates used in our model provide significant predictive value for predicting whether Trump or Clinton wins a county. In this case, because logistic regression relies on fewer assumptions, it produced slightly better results than the other two models.

Next we grew a decision tree and compared its performance with our logistic regression, LDA, and QDA models. We found that the misclassification rates of Trump victories performed well (6% or lower), however, the Clinton win misclassification rates performed poorly (30% or higher). In fact, the misclassification of Clinton victories for the decision trees are actually similar to the other methods before we implemented an optimal threshold. Due to, however, the large misclassification rates of Clinton wins, none of the trees perform as well as the LDA, QDA, and logistic regression models with an optimal threshold.

```
##                  GLM_Pred
## y                     Trump   Clinton
##   Donald Trump    0.9754717 0.0245283
##   Hillary Clinton 0.3176471 0.6823529
```

```
##                  LDA_pred
## class             Donald Trump Hillary Clinton
##   Donald Trump      0.97924528      0.02075472
##   Hillary Clinton   0.35294118      0.64705882
```

```
##                  QDA_pred
## class             Donald Trump Hillary Clinton
##   Donald Trump      0.95660377      0.04339623
##   Hillary Clinton   0.30588235      0.69411765
```

```
##                  ROC_GLM_Pred
## class              Trump Clinton
##   Donald Trump    0.9113  0.0887
##   Hillary Clinton 0.1059  0.8941
##                  ROC_LDA_pred
## class              Trump Clinton
##   Donald Trump    0.8774  0.1226
##   Hillary Clinton 0.0941  0.9059
```

```
##                  ROC_QDA_pred
## class                 Trump   Clinton
##   Donald Trump    0.8981132 0.1018868
##   Hillary Clinton 0.1529412 0.8470588
```

Table 3: Tree Results

tree	misclass_Trump	misclass_Clinton	misclass_Total
small	0.06061	0.2824	0.09135
large	0.07008	0.2941	0.1011
pruned	0.05492	0.2941	0.08809

Results Method 2

Note: We had a split of 60% training and 40% test data for all of test misclassification rates below. We needed a higher percentage of test observations then usually because we were dealing with such a small amount of observations for some clusters.

We see that our boosting model preforms somewhat poorly for K=4 clusters, as it only predicts Trump for 'Cluster 2 of 4 & Cluster 4 of 4'. This is mostly due to the fact that those 2 clusters are almost all made up of counties Trump won. The misclassification rates for K=2 and our base case K=1 are pretty much the same, with the base case performing slightly better. An optimal thresh hold using Youden's statistic felt necessary because our misclassification rates for Clinton without the adjustment were far too high.

Our Logistic Regression model preforms very well for K=4 and K=2 clustering. The clusters are very comparable, if not better than our base case K=1. When we added an optimal thresh hold using the Youden's statistic, we see that our K=4, and K=2 clustering methods were far superior to the base case K=1 with the optimal thresh point.

Misclass Results From Boosting

Table 4: Results From Boosting Method On Cluster (continued below)

Cluster	misclass_Trump	miclass_Clinton	Roc_Trump	Roc_Clinton
1 of 4	0.037	0.347	0.1382	0.1531
2 of 4	0	1	0	1
3 of 4	0.041	0.345	0.1111	0.1505
4 of 4	0	1	0	1
1 of 2	0.008	0.561	0.08083	0.2301
2 of 2	0.013	0.582	0.1056	0.1911
1 of 1	0.009	0.556	0.1251	0.1466

Tot_Misclass	Roc_Misclass	Votes
0.1348	0.1429	623
0.02187	0.02187	1326
0.1353	0.1233	665
0.07237	0.07237	456
0.09428	0.104	1538
0.09661	0.1181	1532
0.09186	0.1283	3070

Misclass results from Logistic Regression

Table 6: Misclassification Rates for Logistic Regression of our clusters (continued below)

Cluster	misclass_Trump	misclass_Hillary	misclass_total
1 of 4	0.05233	0.1558	0.08434
2 of 4	0.009653	0.6667	0.02453
3 of 4	0.09278	0.2535	0.1358
4 of 4	0.005882	0.5833	0.04396
1 of 2	0.04892	0.3077	0.09268
2 of 2	0.01349	0.2258	0.04575
1 of 1	0.0322	0.3333	0.07416

ROC_misclass_Trump	ROC_misclass_Hillary	ROC_misclass_total	Votes
0.1221	0.05195	0.1004	623
0.1081	0.1667	0.1094	1326
0.1186	0.1549	0.1283	665
0.1059	0.1667	0.1099	456
0.1292	0.09615	0.1236	1538
0.02697	0.172	0.04902	1532
0.16	0.09357	0.1508	3070

Datasets

The two raw data sources that our project utilizes are the census and election datasets. The census dataset contains the tract-level 2010 census data that describes the population of each of the tracts in the census. The election dataset contains the votes and the winning candidate for different areas in the US, which is denoted by a fips value which can represent a nationwide, statewide, or countywide area.

The project will merge the two datasets to perform analysis on the election, however the census data contains more high resolution information than the election data since the tracts of the census is more fine-grained the the county-level data of the election dataset. In order to align the two datasets, we aggregated the tract-level census to the county level. We did this by first cleaning up the data by getting rid of unnecessary variables, then we weighted the remaining variables by the population of the county, and finally computing the population-weighted averages of each variable, leaving only data for each county instead of each tract.

The transformation can be seen by looking at the data before and after aggregating.

CensusTract	State	County	TotalPop	Women	White	Citizen
1001020100	Alabama	Autauga	1948	0.5174537988	87.4	0.7715605749
1001020200	Alabama	Autauga	2156	0.508812616	40.4	0.7708719852
1001020300	Alabama	Autauga	2968	0.5404312668	74.5	0.7867250674

State	County	Women	White	Citizen	IncomePerCap
Alabama	Autauga	0.5156733851	75.7882273	0.7374911718	24974.4997
Alabama	Baldwin	0.5115133686	83.10261633	0.7569405651	27316.83516
Alabama	Barbour	0.4617184019	46.23159439	0.7691222338	16824.21643

With the given merged_data set, we had 6142 observations of 30 variables. There were 2 observations per county, one for each of the top two candidates. I wanted to see only the winner of each county, so I sorted the data by decreasing percentage of votes. Then I used the distinct function to keep 1 observation per county with the winning candidate.

Note: I had to used the distinct function on the 'fips' variable instead of the 'county' variable, because multiple counties in the US have the same name but are located in different states.

First 5 variables of first 3 observations for our merged_data

county	fips	candidate	state	votes
roberts	48393	Donald Trump	texas	524
king	48269	Donald Trump	texas	149
grant	31075	Donald Trump	nebraska	367

I then joined our clustered (K=4 & K=2) merged_data with some US map data in order to recreate the map of US counties.

Few variables of first 3 observations for our joined merged_data and US map data (K=4 Map)

long	lat	order	region	subregion
-86.51	32.35	1	alabama	autauga
-86.53	32.35	2	alabama	autauga
-86.55	32.37	3	alabama	autauga

Lastly I joined our clustered merged_data (K=4 & K=2) back with our original merged_data because I needed the covariates from merged_data. I then separated this new dataset by cluster. For example when (K=4) I had four separate data sets, one for each cluster.

Some variables for Cluster 1 & Cluster 2 (Of K=4 Clusters) of this newly created data set.

cluster	county	candidate	total	pct	Women
cluster 1	mcmullen	Donald Trump	497	0.9135	0.4794
cluster 1	hansford	Donald Trump	1947	0.8885	0.4987
cluster 1	bronx	Hillary Clinton	398096	0.8883	0.5317

cluster	county	candidate	total	pct	Women
cluster 2	shackelford	Donald Trump	1502	0.9174	0.5128
cluster 2	wheeler	Donald Trump	2306	0.905	0.4941
cluster 2	winston	Donald Trump	10260	0.8994	0.5078

Discussion

From our four models (logistic regression, LDA, QDA, and decision tree), we found that the logistic regression model performed the best with an optimal threshold set. Without the optimal threshold, all the models suffered from the same problem – high misclassification rates. Although we found the logistic regression model to perform the best with an optimal threshold set, it only performed slightly better than the QDA and LDA models with optimal thresholds.

From our clustering method, we found that clustering before performing supervised learning returns mixed results. With our boosting method, the base case (K=1) preformed better than both our clusters (K=2 & K=4). However when looking at our logistic regression model we see that clustering before performing predictions seemed to give us slightly superior results. One of the main reasons our cluster model seemed to struggle with predictions was due to the lack of observations in some clusters. Due to this I had to use 60% training and 40% test data. However, even this didn't improve our results with the boosting method. One of the main reasons I believe clustering failed with the boosting method is because of the lack of Clinton won counties in some clusters. For example Cluster 2 of 4 and 4 of 4, had barely any Clinton won counties

To be honest, clustering wasn't too great for this project because Trump won around 5x more counties than Clinton. Therefore almost every cluster created was gonna most likely predict Trump over Clinton.

A final problem we ran into is that we had to include the Dataset section near the end of our report, because we made special manipulations with clusters near the end of our code to the datasets, it just made more sense and was easier to include it near the end of the project.