Final Project

Tristan Chen & Elias Parzen

6/2/2021

Problem 1 *What makes voter behavior prediction (and thus election forecasting) a hard problem?*

As the 2016 presidential race demonstrated, election forecasting is a far-from-trivial task. Legions of highly educated, intelligent experts feeding vast troves of data into state-of-the-art statistical models had their reputations forever tarnished by confidently predicting an easy win for Clinton, when in reality Trump triumphed with a comfortable lead over his opponent in the electoral college. In fairness to the pollsters and pundits, it's simply impossible to eliminate error when predicting the outcome of an event as complex as a presidential election. With literally hundreds of millions of eligible voters and a poll's sample size generally being several orders of magnitude smaller than that, there will always be some amount of variance in poll results due to sampling errors. Furthermore, seemingly trivial aspects of how a poll is designed, like the exact wording of the questions, or even the medium the poll is conducted through can bias the results; the famous 1936 Literary Digest Poll debacle was a result of phone owners voting differently than those without phones, which caused the Digest's phone poll to suffer from extreme sampling bias.

Even with an unbiased sample, respondents aren't always honest about their intention. The "Shy Tory Effect" which plagues British polls is a result of Conservative supporters being less comfortable revealing their party preference to pollsters than Labour voters are.

When you add in factors such as shifting demographics, the difficulties of predicting voter turnout, and general statistical noise, it becomes readily apparent why election forecasting is such a hard problem. There are too many ways for error to creep into polling results, which means that if too many of those errors are all in the same direction, election model predictions can all-too-easily end up at odds with reality, as was the case in 2016.

Problem 2 *What was unique to Nate Silver's approach in 2012 that allowed him to achieve good predictions?* Nate silver is a statistician, writer, and founder of the data journalism website, FiveThirtyEight. While he first gained noteriety by developing a well-regarded model for predicting the performance of professional baseball players, Silver became a household name after he correctly predicted the results of the 2008 presidential election in 49 out of 50 states in 2008. In the 2012 election, he beat his own record by accurately forecasting the outcome of all 50 states.

Silver's model, which he and his team have continuously iterated upon, makes prediction based on not just polling data, but also electoral history and demographics. By aggregating polls and weighing them against other features, the model is more resilient to polling error than those relying on individual polls.

While FiveThirtyEight gave Hillary Clinton the edge going into the 2016 election, Silver and co. did assign the future President Trump a much higher probability of victory than did many of their competitors.

Problem 3 *What went wrong in 2016? What do you think should be done to make future predictions better?* As was previously explored, polls, and therefore models based on polls, suffer from many sources of error. The poor performance of 2016 election forecasts was in large part due to a perfect storm of polling errors: while the errors were not incredibly large, they were generally biased in the same direction (underestimating Trump's support), and not equally distributed, but rather concentrated in key states. Some version of the "Shy Tory Effect" may have been at play, with at least one polling firm noting a discrepancy in results between voters responding to pre-recorded questions and those being interviewed with live, potentially judgmental

interviewers. Turnout for Democrats was lower than expected, especially in the Midwest, and there may have been a last-minute surge in Trump support as undecided voters made up their minds at the eleventh hour.

While President Trump lost his reelection bid in 2020, many polls again underestimated his popularity, leading to a much closer race than expected. This continued, systemic polling error is an indication that there are considerable, industry-wide problems that need to be addressed. Sampling is likely one source of error that must be addressed; clearly the proportion of respondents favoring Trump in the sample was not reflective of their proportion of the electorate in several states. Clearly, conventional polling approaches are not up to the task of getting certain types of voters to participate or feel comfortable honestly communicating their intentions. Advances must also be made in more reliably predicting voter turnout and enthusiasm, as voters' intentions don't affect elections unless they actually show up to cast their vote.

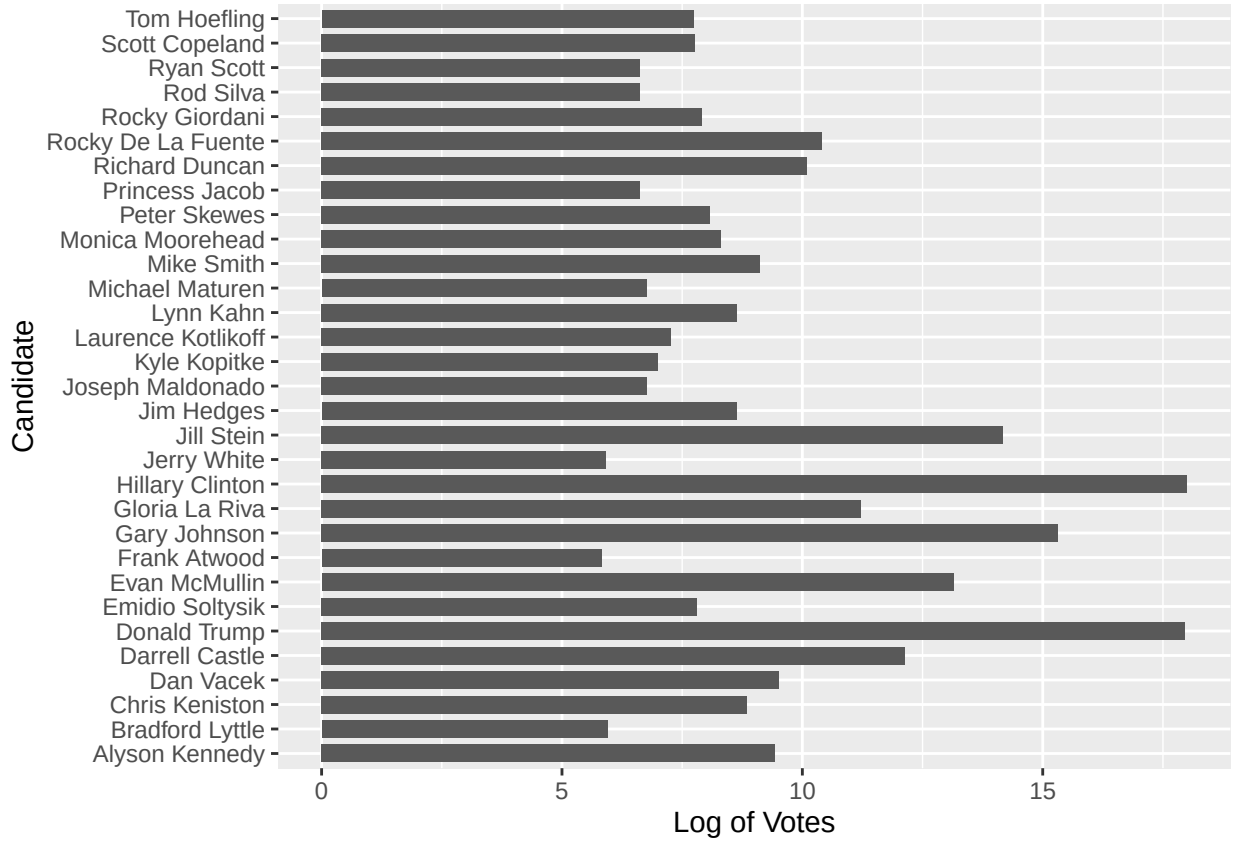
As election forecasters lick their wounds and recalibrate their models, it's also important that they don't overcorrect as a result of their bruised egos. It's certainly possible that there were some polling challenges specific to the 2016 election that won't extend to future presidential races. The former president was a unique candidate, and he may have had unique effects on turnout, demographic support, and attitudes towards pollsters. It would be a mistake to throw the baby out with the bathwater by making drastic changes to a model that may have worked properly had someone else won the Republican primary.

Problem 4

The dataset has 18345 rows and 5 columns after removing rows with fips = 2000. This data is removed because data with fips = 2000 represents summary data for Arkansas, but state summary data is represented with the names of states as fips values. The data with fips = 2000 is a duplicate of the data with fips = AK.

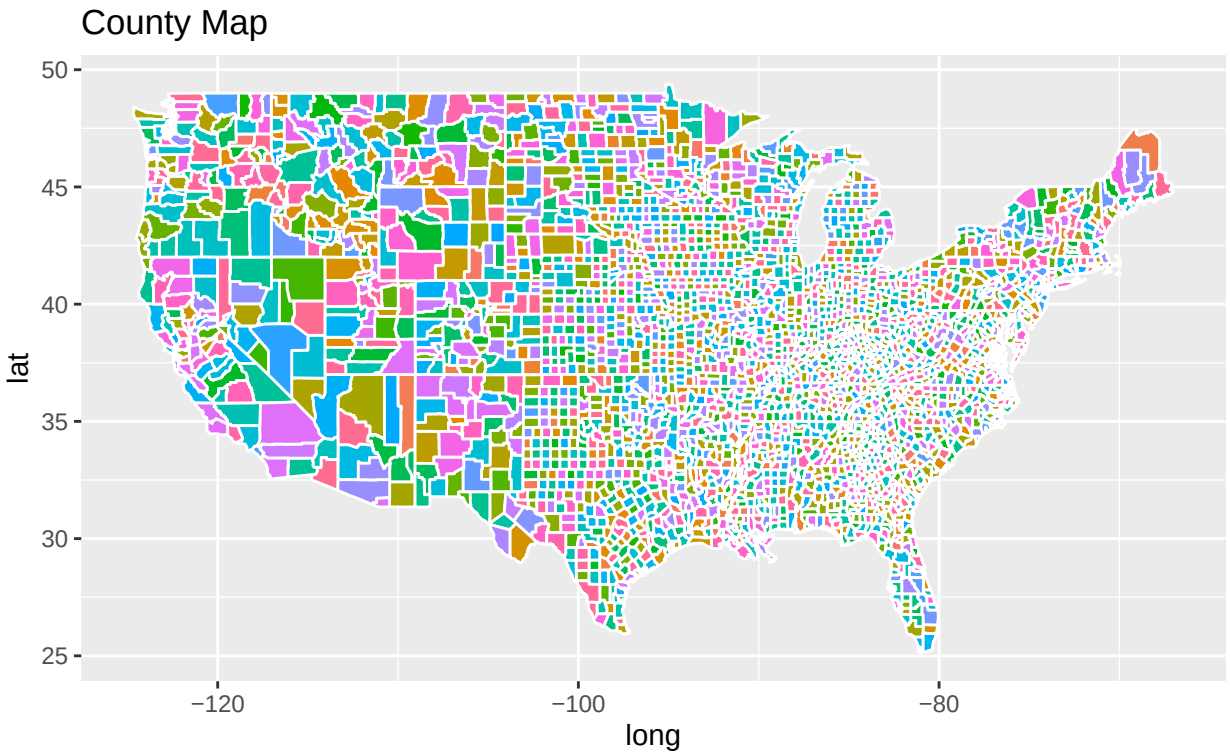
Problem 5

Problem 6 There were 31 named candidates in the 2016 election.

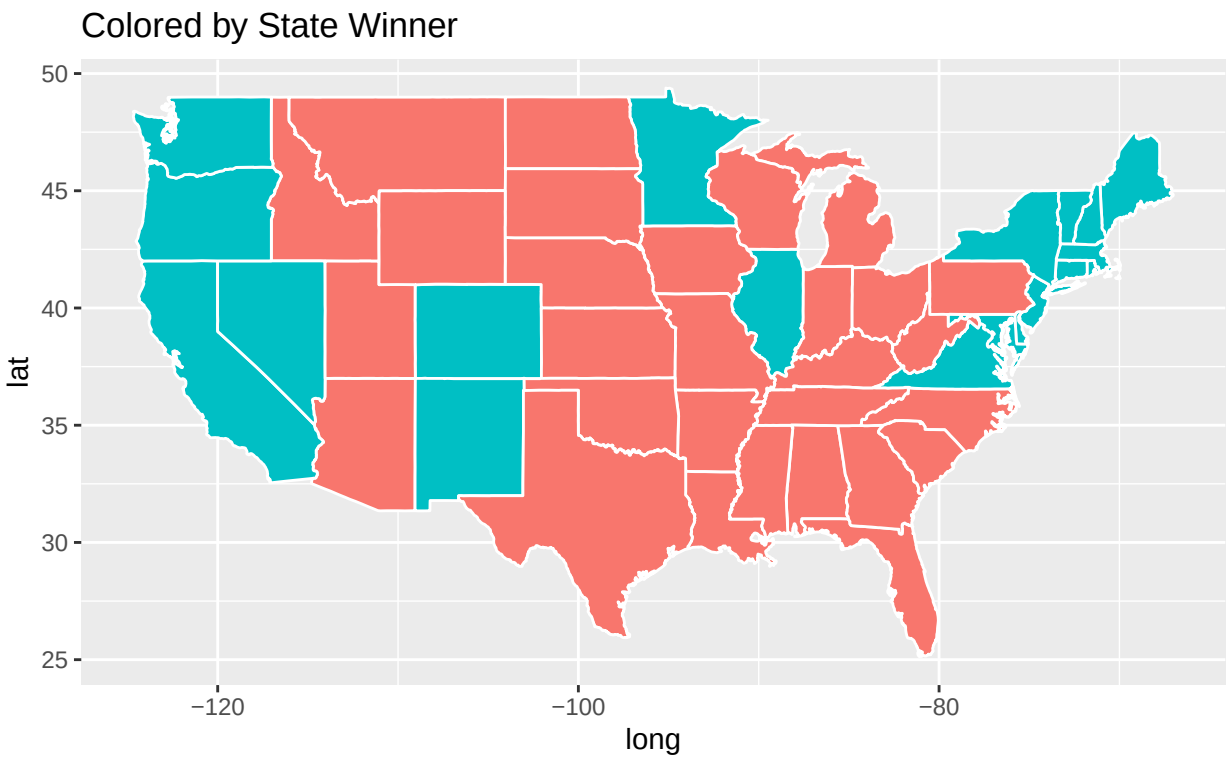


Problem 7

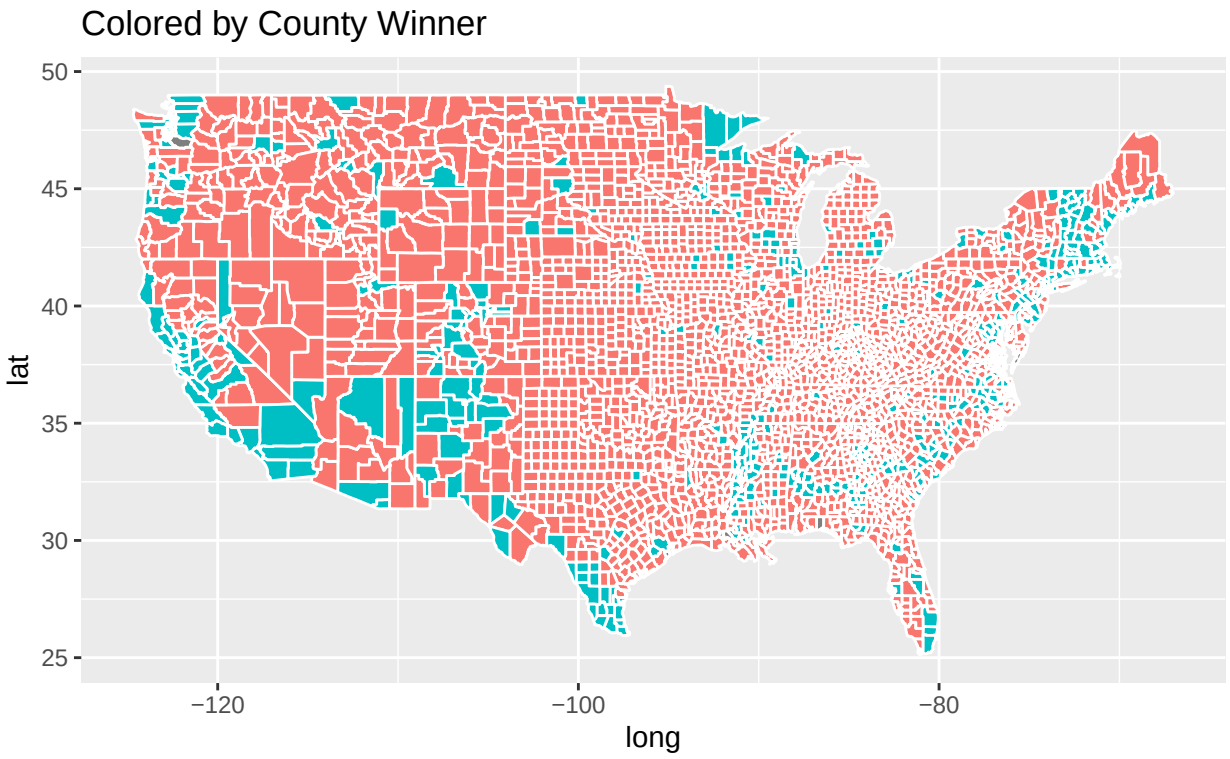
Problem 8



Problem 9



Problem 10

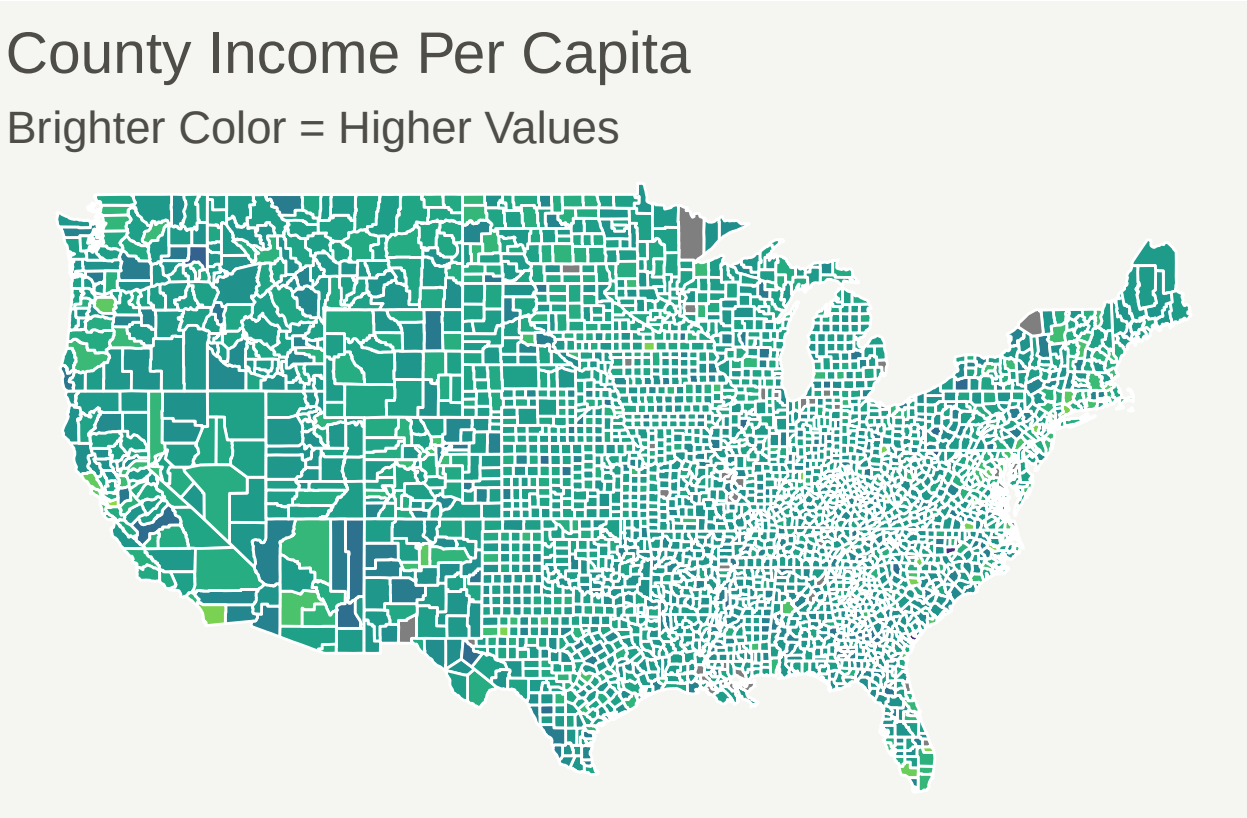


Problem 11

Since there was much talk of how the impact of income on voter preferences had/hadn't changed in the 2016 election, we thought it would be interesting to create a choropleth county map to compare income levels to the previous county election map. To avoid merely capturing the impact of population size, we used Income Per Capita rather than Income as our numerical measurement of wealth.

Table 1: Example rows of census.ct

State	County	Men	White	Citizen	Income	IncomeErr	IncomePerCap	IncomePerCapErr	I
Alabama	Autauga	48.43266	75.78823	73.74912	51696.29	7771.009	24974.50	3433.674	12
Alabama	Baldwin	48.84866	83.10262	75.69406	51074.36	8745.050	27316.84	3803.718	13
Alabama	Barbour	53.82816	46.23159	76.91222	32959.30	6031.065	16824.22	2430.189	26
Alabama	Bibb	53.41090	74.49989	77.39781	38886.63	5662.358	18430.99	3073.599	16
Alabama	Blount	49.40565	87.85385	73.37550	46237.97	8695.786	20532.27	2052.055	16



Problem 12

Problem 13

We chose to center and scale the features before running PCA in this instance. Some of our variables, e.g. Minority, are percentages capped at 100, whereas variables like Income are measured in the tens of thousands. This disparity would lead to the bigger features drowning out the effect of the smaller features by size alone in PCA. By centering and scaling the variables, we can avoid this problem and make sure the output of PCA reflects the importance of each variable rather than just the size.

In ct.pc, the three features of PC1 with the largest absolute values are Income Per Capita, Child Poverty, and Poverty. In subct.pc, those features are Income Per Capita, Professional, and Poverty.

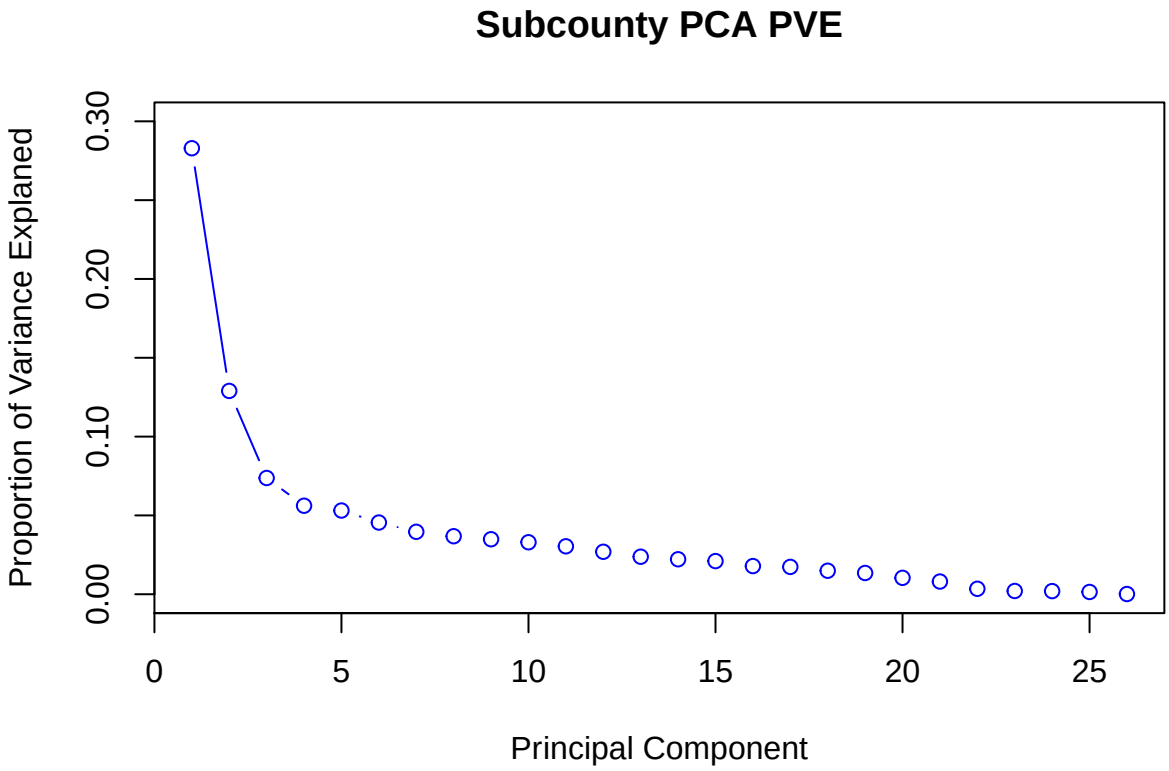
In ct.pc, Income Per Capita has a positive coefficient, while Child Poverty and Poverty are negative..

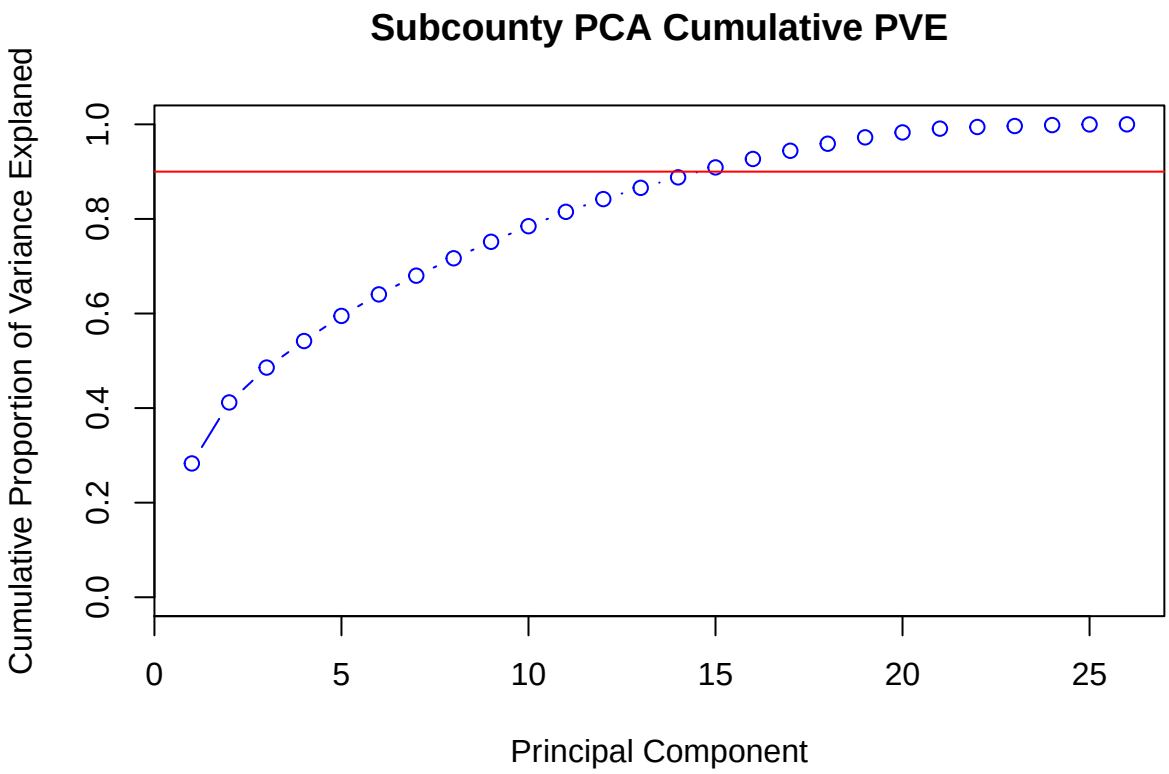
In subct.pc, Income Per Capita and Professional were negative, while Poverty was positive.

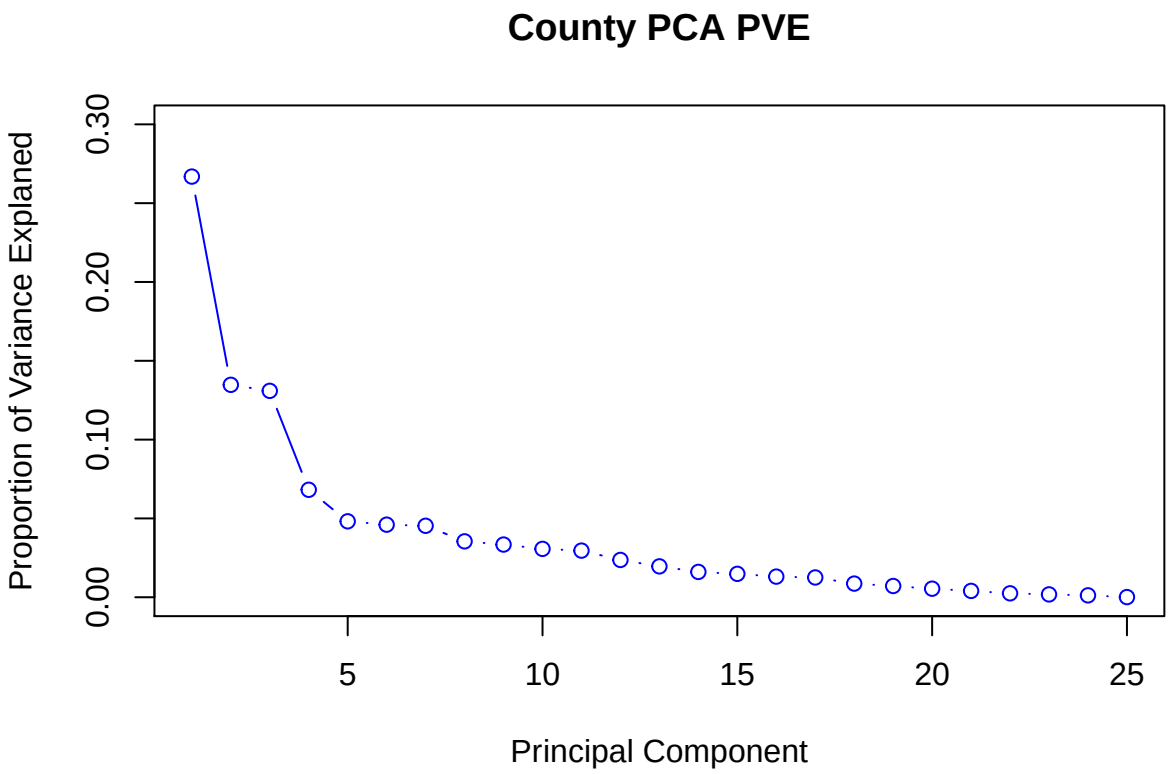
When features have opposite signs, it means that they are inversely correlated. In the context of PCA, the higher the values of these features an observation has, the further apart those observations will be on that

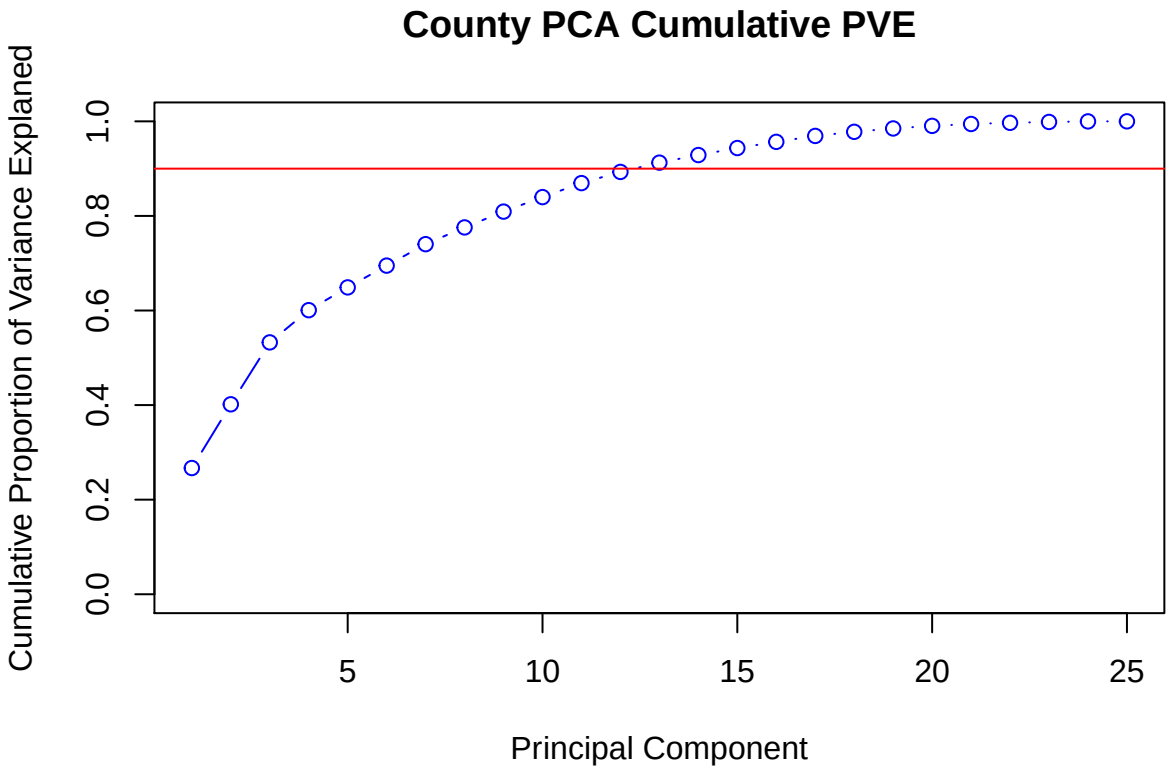
principal component axis.

Problem 14




Subcounty PCA Cumulative PVE
Cumulative Proportion of Variance Explaned
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
Principal Component


County PCA PVE
Proportion of Variance Explaned
0.00
0.10
0.20
0.30
5
10
15
20
25
Principal Component



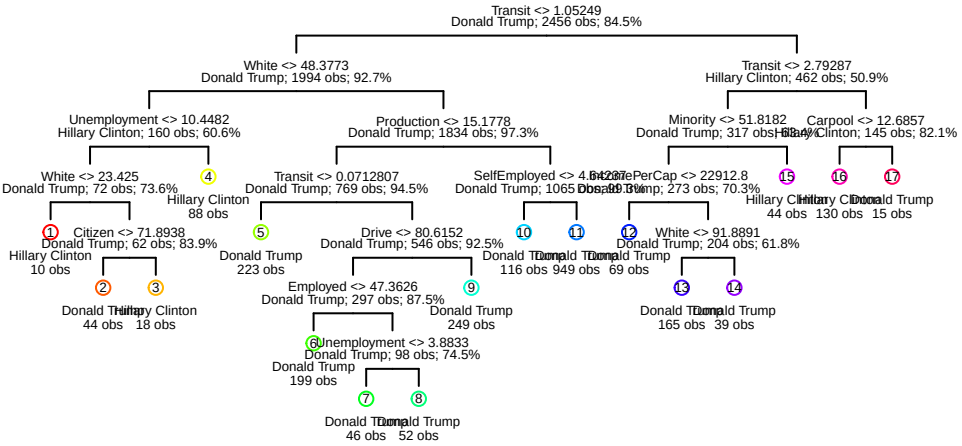
For subct.pc, 15 principal components are needed to capture 90% of the variance. For ct.pc, only 13 principle components are required to meet the same benchmark.

Problem 15

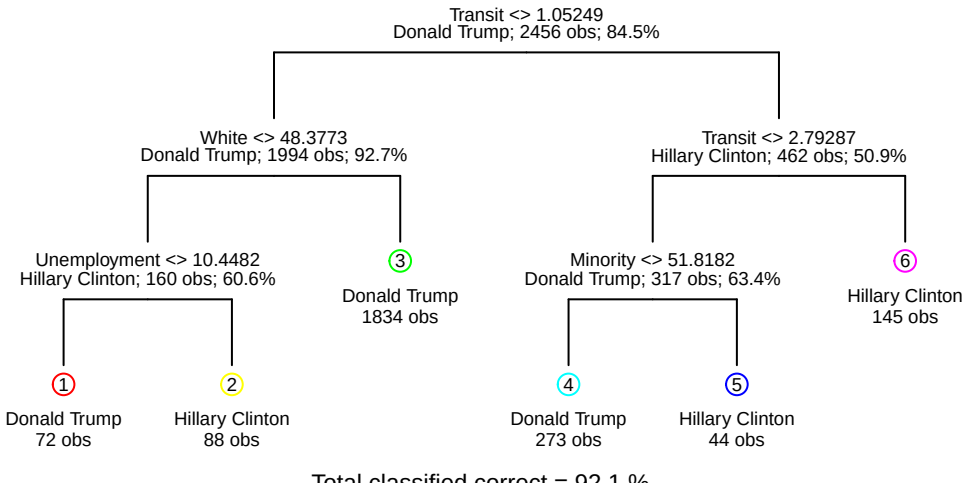
Our pre-PCA hierarchical clustering places San Mateo County in cluster 9, along with other high-income counties such as Santa Clara, Marin, and Douglas. This cluster is one of the smallest of the 10. Since the data in census.ct is unscaled, this result may be the outcome of the unscaled income variables swamping out the influence of other features

When we instead cluster based on the first 5 principal components, San Mateo is placed in the much larger cluster 4. This seems to be a better fit, as the other members of this cluster are similar to San Mateo across a variety of variables, not just the income-related ones.

Problem 16



Total classified correct = 92.8 %



From the pruned tree, we can see that the most important factors that affect county candidate choices are transit rate, rate of white people in a county, and unemployment rate.

If the transit rate for a county is less than 1.05% and if the county is more than 48.3773% white, then there is a 92.79% chance that Donald Trump will take that county. If the county is less than 48.3773% white, and if unemployment rate is higher than 10.45%, then Hillary Clinton wins 88 to 72.

If the transit rate for a county is greater than 2.79% , then Clinton has a 50.9% of winning that county. If the transit rate is less than 2.79% and the county is less than 51.82% minorities, then Donald Trump wins 273 to 44.

Problem 17

Under the significance threshold of 0.01, White, Citizen, Income, IncomePerCap, Professional, Service, Production, Drive, Carpool, Employed, PrivateWork, and Unemployment are significant variables. This is not consistent with what we saw in decision tree analysis, where we previously saw that Transit and Minority were also significant variables.

We see that the Citizen coefficient is 0.1274, meaning that with all other variables fixed, a one unit increase in citizenship rate in a county corresponds in a multiplicative increase in the odds of Clinton winning by exp(0.1274), or 1.136. We also see that the Unemployment coefficient is 0.210, meaning that with all other variables fixed, a one unit increase in unemployment rate in a county corresponds to a multiplicative increase in the odds of Clinton winning by exp(0.210), or 1.23.

Problem 18

The optimal value of λ in cross validation is 0.0005. For this value of lambda, the non-zero coefficients are Men, White, Citizen, Income, IncomeErr, IncomePerCap, IncomePerCapErr, Poverty, ChildPoverty, Professional, Service, Office, Production, Drive, Carpool, Transit, OtherTransp, WorkAtHome, MeanCommute, Employed, PrivateWork, SelfEmployed, FamilyWork, Unemployment. There is only one zero coefficient, which is Minority,

Table 2: LASSO regression coefficients

	Coefficient
(Intercept)	-26.9133120
Men	0.0700545
White	-0.1308036
Citizen	0.1316442
Income	-0.0000607
IncomeErr	-0.0000161
IncomePerCap	0.0002034
IncomePerCapErr	-0.0002564
Poverty	0.0323020
ChildPoverty	-0.0015493
Professional	0.2605642
Service	0.2988803
Office	0.0617987
Production	0.1379091
Drive	-0.1810180
Carpool	-0.1438854
Transit	0.1065950
OtherTransp	-0.0180684
WorkAtHome	-0.1218180
MeanCommute	0.0408852
Employed	0.1944336
PrivateWork	0.0977514
SelfEmployed	0.0009233
FamilyWork	-0.7439136
Unemployment	0.1972567
Minority	0.0000000

Table 3: Training error and test error records

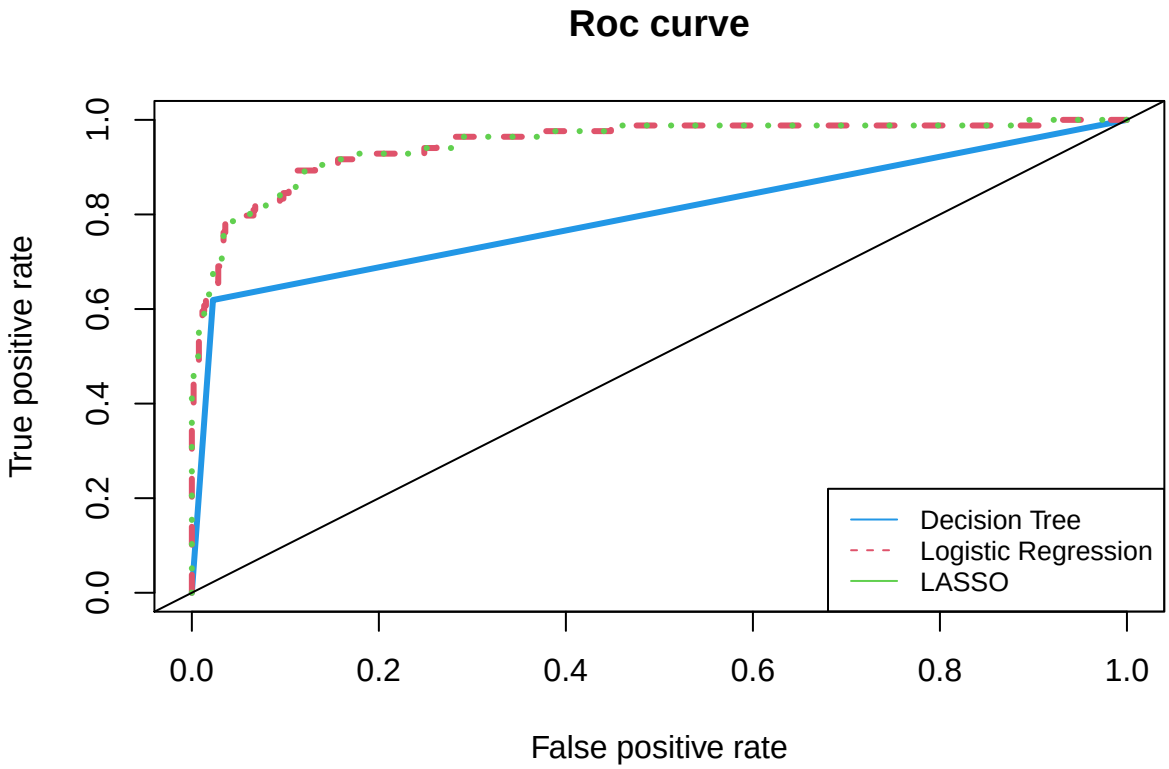
	train.error	test.error
tree	0.0789902	0.0715447
logistic	0.0696254	0.0634146
lasso	0.0692182	0.0666667

Table 4: AUC records

	AUC
tree	0.7982244
logistic	0.9481212
lasso	0.9490629

indicating that the model is not sparse. Compared to unpenalized logistic regression, the test error for LASSO regression is slightly higher.

Problem 19



Based on the results of decision tree classification, we see that it is easy to implement and interpret. The cons are that decision trees have a tendency to overfit, meaning that small changes in the data can lead to completely different classifications. This method is also less accurate in its predictions compared to other models. The logistic regression model allows us to see the importance of each variable and is useful for binary classification. The cons are that logistic regression does not work well with nonlinear or small datasets. LASSO regression is useful when there are many irrelevant predictors, and it allows us to select the important variables. The cons are that the model will be biased towards the selected variables, and that it does not affect the variance much if most of the predictors are relevant.

Problem 20

Use any tools at your disposal to make your case: visualize errors on the map, discuss what does/doesn't seems reasonable based on your understanding of these methods, propose possible directions (collecting additional data, domain knowledge, etc).

Table 5: Training error and test error records

	train.error	test.error
tree	0.0789902	0.0715447
logistic	0.0696254	0.0634146
lasso	0.0692182	0.0666667
knn	0.0594463	0.0813008

In this project, we predicted election results using several classification methods, including decision trees, logistic regression, and LASSO regression. Based on the decision tree model, we determined that Transit was the most important predictor, followed by White, Minority, and Unemployment. This differs from the significant variables from our logistic regression model, which includes Citizen, IncomePerCap, Professional, Service, Production, Drive, Employed, PrivateWork, and Unemployment, but not Transit or Minority. By looking at the AUC for each of the 3 classification methods, we can see the differences in their effectiveness. The decision tree method has a much lower AUC value, 0.798, than logistic regression or LASSO regression, meaning that it is not as effective in modeling the election data. This may indicate that the election data does not fit well into rectangular regions. The test error for this model was also higher than the other 2 models, possibly illustrating the tendency for decision trees to overfit, as small changes to the dataset caused by polling errors may have completely changed the model. The logistic and LASSO regression models appear to be much more accurate in its predictions, with AUC values of 0.948 and 0.949 respectively. The logistic regression model fits the data well, as this dataset involves binary classification. The LASSO regression model does not appear to be a significant improvement over the unpenalized log regression model, as its AUC value is only slightly higher and its test error is actually slightly higher. This indicates that the true model is not sparse, as most of its predictors are relevant. This is further proved by the fact that only one of the predictors, Minority, was reduced to zero. These models may be improved if the dataset included polling and demographic data from previous years, similar to Nick Silver's model. This would be useful in reducing biases and improving each model's resilience to errors.

Interesting Questions

We explored modeling the election data using logistic regression and a decision tree, but we were curious about how a K Nearest Neighbors model would perform. As a non-parametric model, KNN makes no assumptions about the structure of the data, unlike logistic regression, which assumes that the log odds of an observation falling into a particular class can be written as a linear combination of the predictors. While decision trees are non-parametric, they work best when the decision boundary of our classification problem is rectangular; KNN is much more flexible in terms of the shape of the decision boundary. By trying out a KNN model, we can see if the assumptions implicit in the other two models hold up to scrutiny.

As we can see from these results, although the training error for the KNN model was markedly lower than those of the other three, the test error was actually worse. This indicates that the extra flexibility that the KNN model offers was in this case a liability rather than an advantage. A lower training error combined with a larger test error is an indicator that the model was overfit, even with an optimal k chosen through cross validation.

Another question we were curious about was dimension reduction for our logistic model. We already used a LASSO model that set the coefficients of non-significant variables to 0, effectively removing them from the model. What if instead we trained our logistic model on the output of a PCA on the data?

Table 6: Training error and test error records

	train.error	test.error
tree	0.0789902	0.0715447
logistic	0.0696254	0.0634146
lasso	0.0692182	0.0666667
knn	0.0594463	0.0813008
PCA logistic	0.0883550	0.2308943

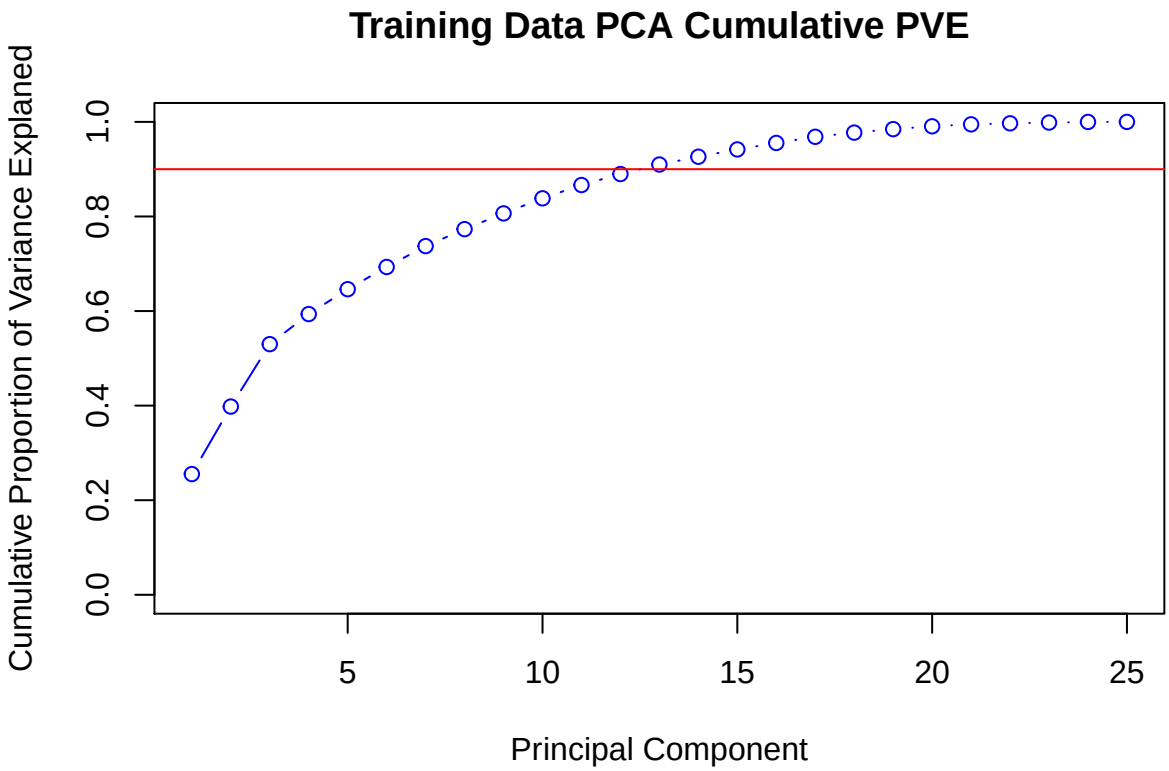
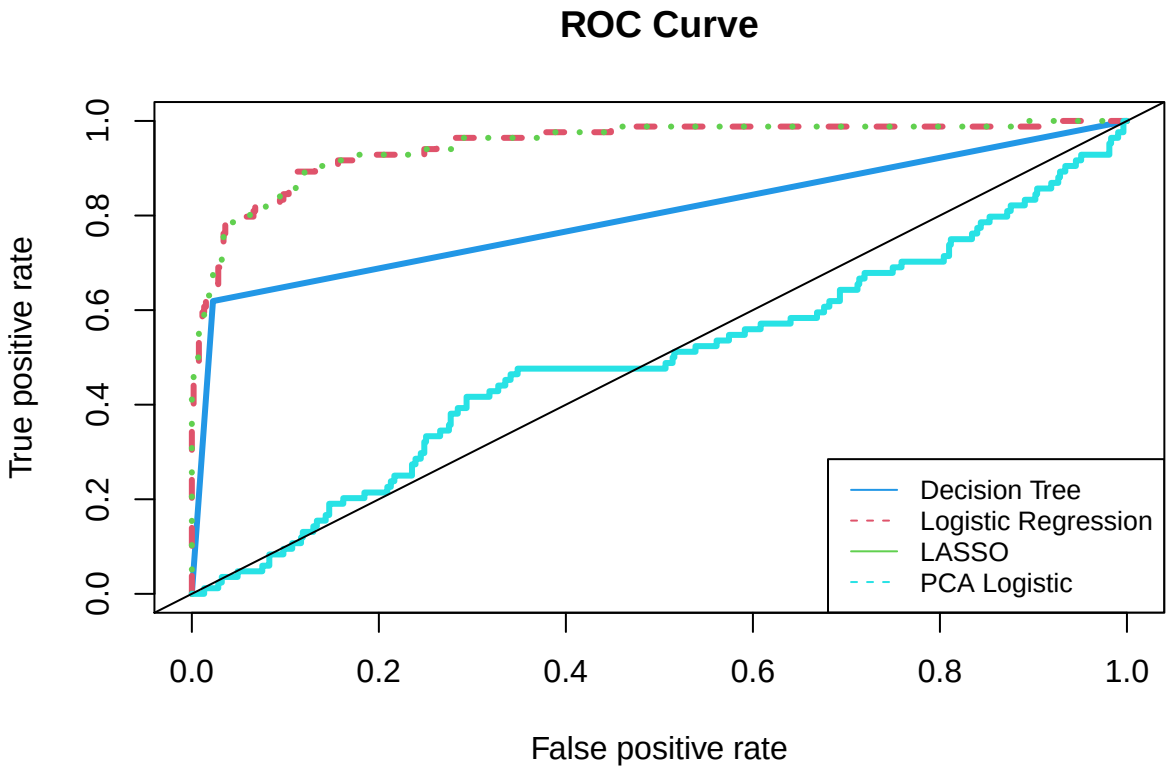


Table 7: AUC records

	AUC
tree	0.7982244
logistic	0.9481212
lasso	0.9490629
pca logistic	0.4937001



As can be seen from the ROC curve and AUC table, logistic regression, when performed on the PCA output of the training data, severely underperforms every other model. One explanation for this could be that the PCA output of the test data is significantly different from that of the training data. Indeed, an examination of the rotation matrices reveals many differences in the coefficients of the features.

One remedy for this would be to perform PCA on the entire original dataset, before the train/test split. This would reduce the likelihood of big discrepancies between the training and testing PCA and potentially boost the accuracy.