STAT 4221 Final Project

Carlyle Morgan

5/4/2022

Research Question

Can we use demographic data from the ACS to determine if a given zipcode had an increase or decrease in rental prices during the first months of the COVID-19 Pandemic (as measured by the ZORI Index)?

Pre-Processing and Exploratory Data Analysis

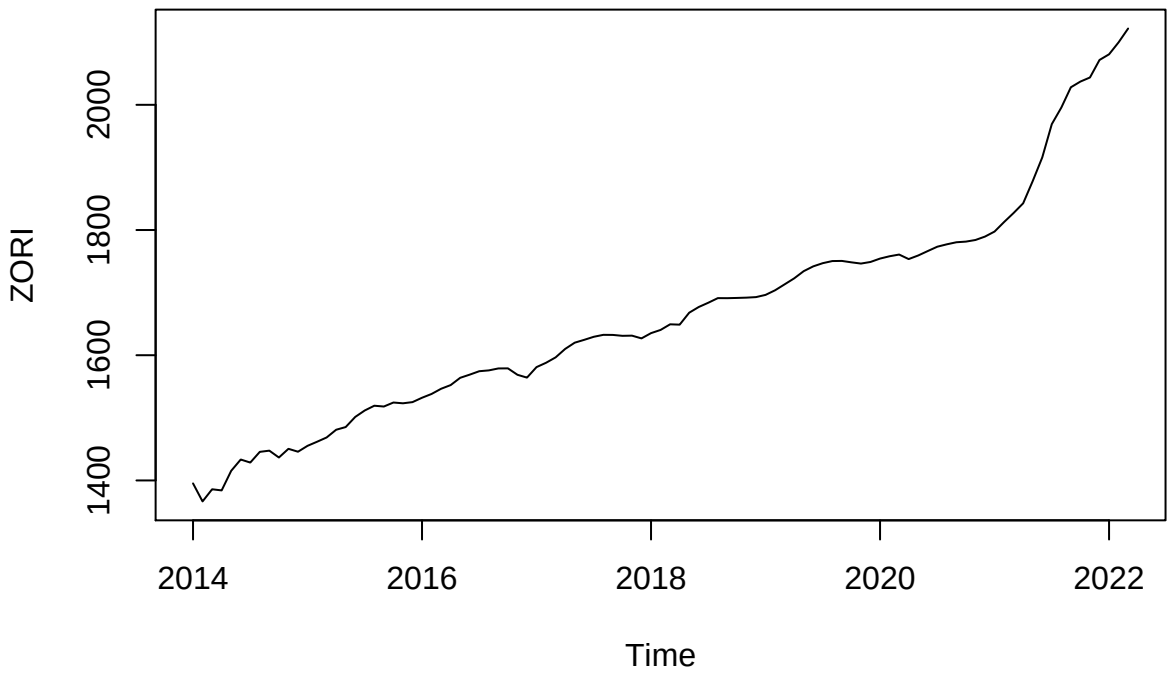
ZORI

The Zillow Observed Rent Index (ZORI) measures changes in asking rents over time, controlling for changes in the quality of the available rental stock.

I downloaded ZORI data from April of 2014 to March of 2022 at the zipcode level. There were 2249 zipcodes for which ZORI data was available. ZORI consists of monthly time-series data, so it is helpful to visualize general trends over the observed period.

Mean ZORI over the period:

```
monthmeans<-apply(zori[,5:103],2,function(x) mean(x, na.rm=TRUE))
ts_mme<-ts(monthmeans,start=c(2014,1), end=c(2022,3), frequency=12)
plot(ts_mme, ylab = "ZORI")
```



There appears to be a yearly seaonality across the dataset.

Now let's investigate COVID's impact on mean rents:

```
covidmeans<-apply(zori[,76:88],2,function(x) mean(x, na.rm=TRUE))
ts_covid<-ts(covidmeans,start=c(2019,12), end=c(2020,12), frequency=12)
ts_covid
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2019                                                                       
## 2020 1754.387 1758.084 1760.808 1753.587 1759.345 1766.414 1773.319 1777.176
##           Sep      Oct      Nov      Dec
## 2019                           1748.956
## 2020 1780.472 1781.490 1784.064 1789.510
```

There was a distinct dip in rents between March of 2020 and April of 2020. Let's add feature 'covidchange' with value 1 if rents increased or remained the same and 0 if rents decreased in a certain zipcode. Of the 2226 zipcodes in the dataset, 1252 had decreased rent over this period while 974 had increased or stagnant rents.

ACS Data

The American Community Survey is sort of like a miniature census. It collects a wide variety of demographic data. I used data from the 2014 American Community Survey at the zipcode level. We will only want to look at data for which we have both ZORI and ACS data, but the latter set completely encompasses the first. The ACS data primarily consists of population counts on the basis of age, gender and race.

The ACS by its very design will have several linearly dependent features, leading to problems of rank deficiency if the set is used for linear regression. By removing linear dependent features, working with the dataset will also be more computationally efficient. I will cheat and use the lm() function's built-in ability to detect linearly dependent columns to filter such values out of the ACS. We also don't care about regression on columns that predict margin of error for demographics (Ex. we estimate the number of children under 5 with X% error).

The 2014 ACS Data originally had 33121 observations of 326 features. After filtering observations that do not have a match in the ZORI dataset, the number of observations decreased to 2246. After condensing linear dependent columns and removing columns that provided relative or absolute margins of errors for ACS estimates, 326 features reduced to 61.

Normalization of the Feature Space

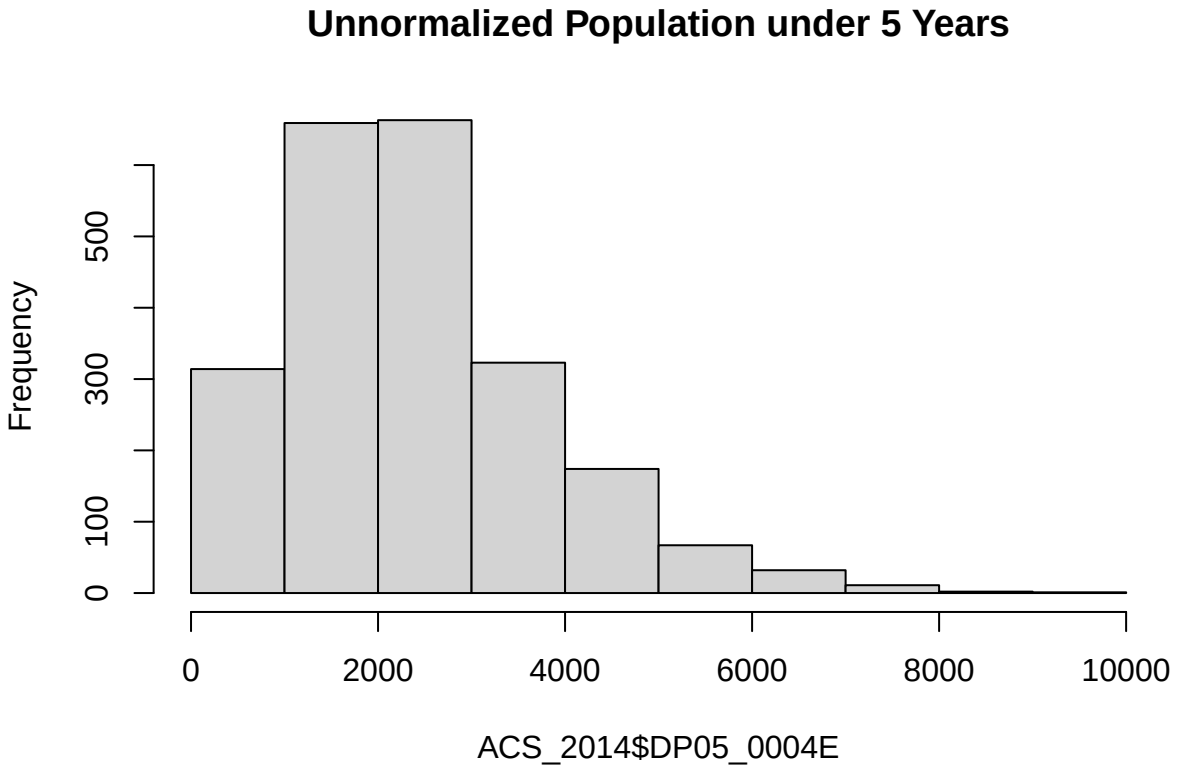
Given the difference in orders of magnitude between features, I decided that the data needed some form of normalization. Since most features are discrete demographic data, they take only positive real values. Therefore one of the following normalization seemed appropriate.

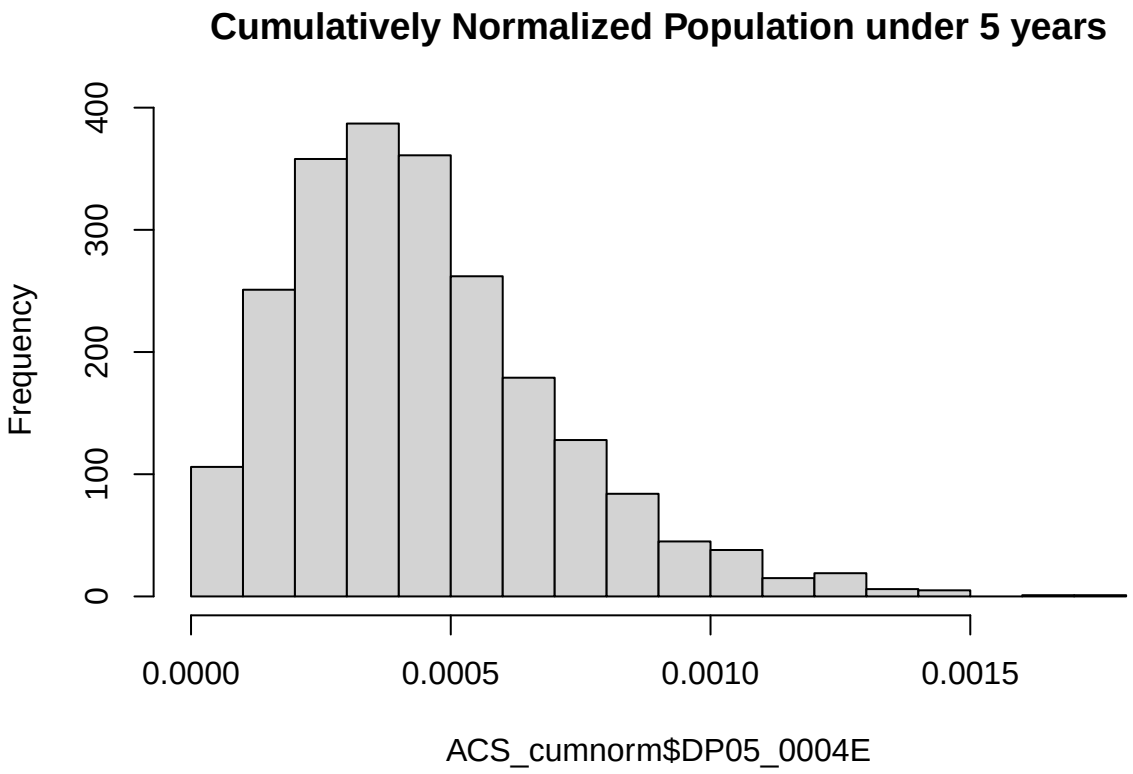
Cumulative Norm

For observation X_i of feature j, X_{ij}, the normalized observation would be:

$$\tilde{X}_{ij} = \frac{X_{ij}}{\sum_k X_{kj}}$$

. This is equivalent to dividing the individual values of the features by the sum of each features, with the new normalized values between 0 and 1.





For these histograms of the "Under 5 Years population" in zipcodes in the ZORI dataset, the cumulatively normalized population seems to emulate the distribution of the unnormalized population.

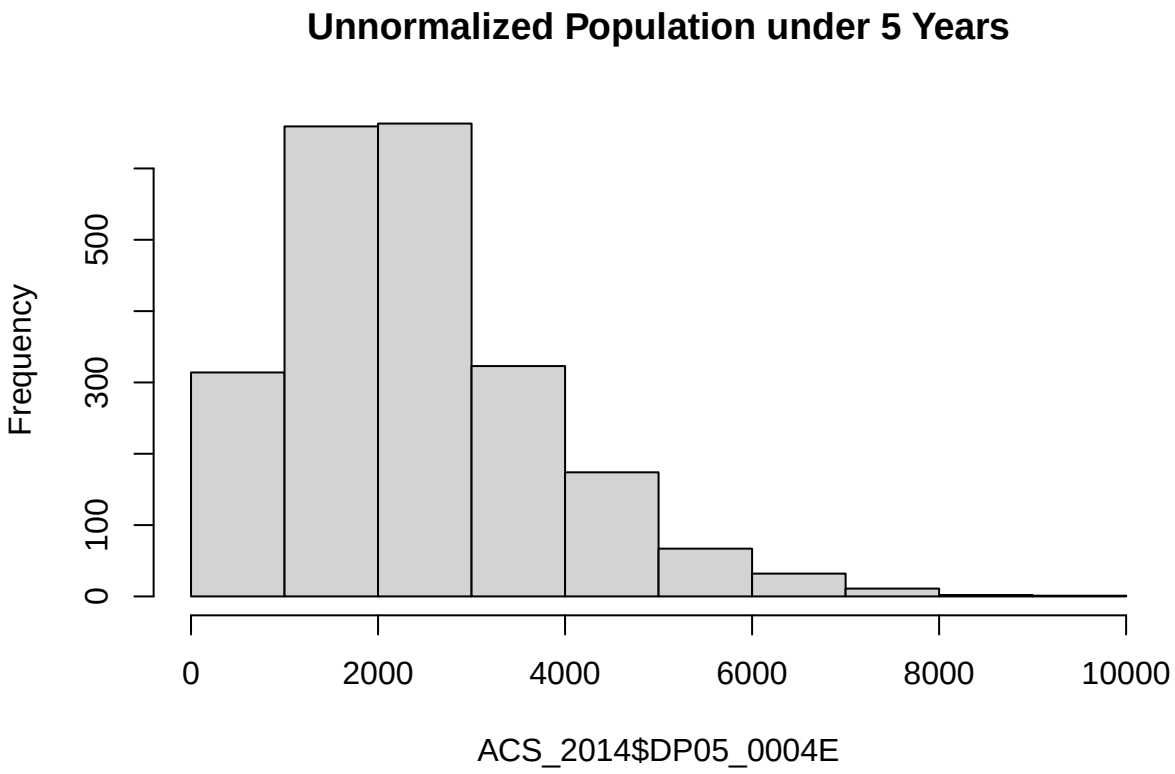
Maximum Norm

For observation X_i of feature j, X_{ij}, the normalized observation would be:

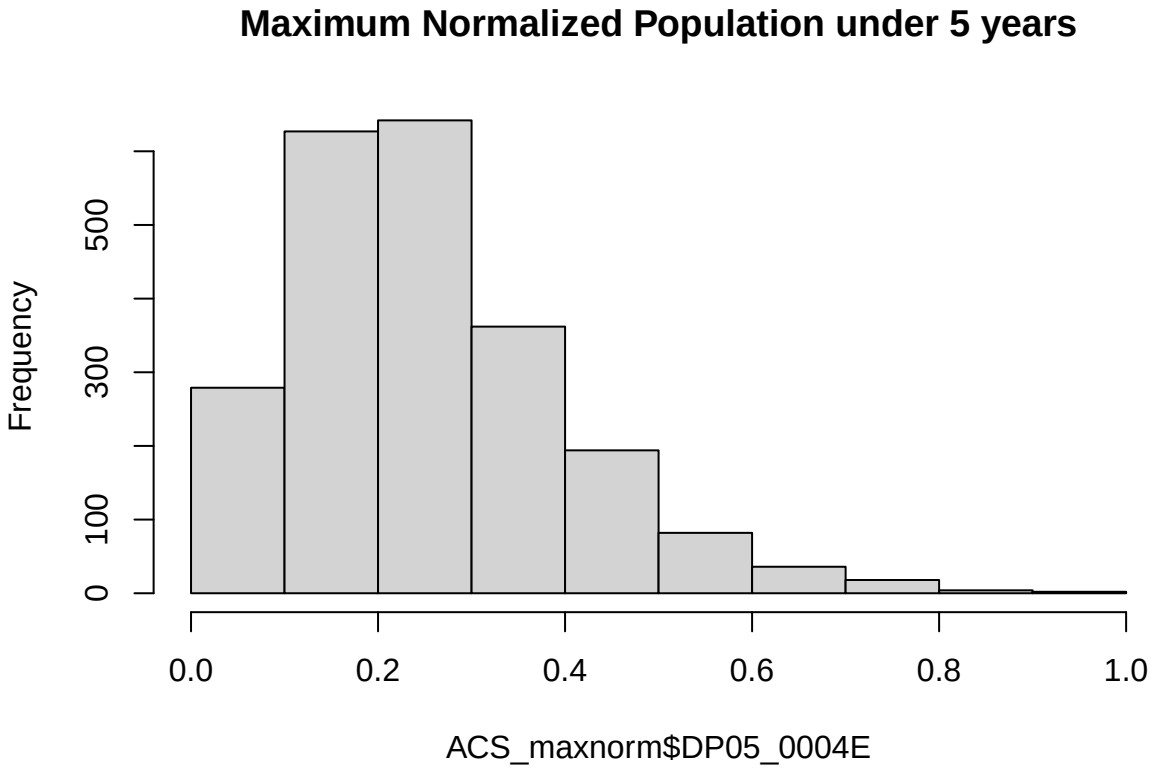
$$\tilde{X}_{ij} = \frac{X_{ij}}{\max_k\{X_{kj}\}}$$

. This is equivalent to dividing the individual values of the features by the max of each features, with the new normalized values between 0 and 1.

```
ACS_maxnorm<-data.frame(apply(ACS_2014, 2, function(x) x/max(x)))
hist(ACS_2014$DP05_0004E, main = "Unnormalized Population under 5 Years")
```




```
hist(ACS_maxnorm$DP05_0004E, main="Maximum Normalized Population under 5 years")
```



For these histograms of the "Under 5 Years population" in zipcodes in the ZORI dataset, the maximum normalized population seems to emulate the distribution of the unnormalized population.

Which to use?

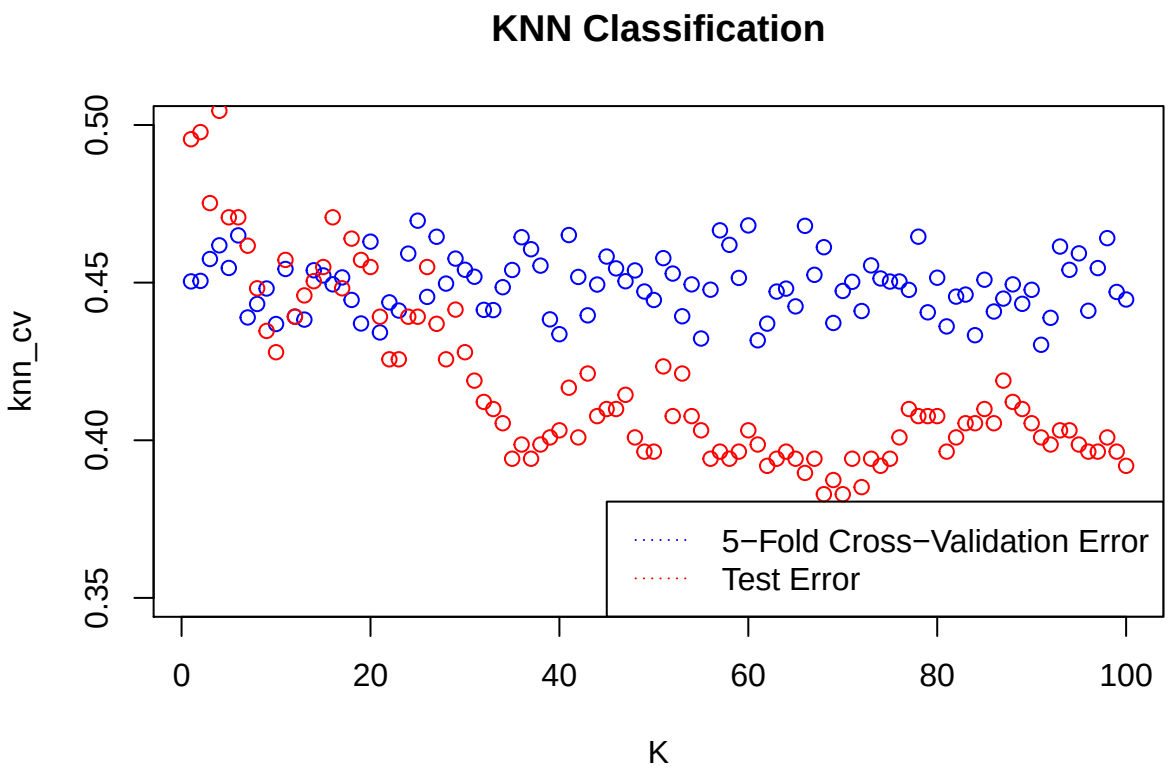
Both norms seem to do the desired task of mapping the existing values of the features to values between 0 and 1 and roughly preserving the distribution of the unnormalized values. However, the maximum norm seems to do a better job with producing values that are not extraordinarily small, making greedy algorithms I might use in my analysis a bit easier to program. The cumulative norm also creates incredible sparse data concentrated close to zero, making decisions stumps also more difficult to program and utilize. Therefore, I chose to use the **max norm**.

Classification

I chose to use a variety of different classification algorithms. To standardize results across models, I created a test set composed of 20% of the original data that was used to calculate the test error across algorithms. The test set had 174 zipcodes where rents increased over the first months of the pandemic and 270 where rents did not. The training set had 799 zipcodes where rents increased and 980 where rents did not. Where cross validation for hyperturning was performed, cross validation sets across models were not standardized.

The models I decided to end up implementing wre K-Nearest Neighbors, Logistic Regression, LDA, QDA, and Adaptive Boosting. My initial hypothesis was that of these, Adaptive Boosting would be the most powerful, especially with such a large number of features. I thought that Adaptive Boosting would do a good job of performing feature selection while also making quality predictions.

K-Nearest Neighbors



It seems that from this graph, a solid choice for K would between 40 and 60. Oddly enough, cross validation error seems relatively constant across the parameter space. This indicates that one should be a bit skeptical about the

observed test error, especially when it is drastically below the cross-validation error. The overall trend is clear though: we should probably use at least more than 10 neighbors in the classification model.

Logistic Regression

From the regression summary, it appears that the following ACS survey entries had significant decision power: 29, 62, 66, and 72. This corresponds to the following entries: total population that is one race only, Asian population, Hispanic or Latino population (of any race), and Non-Hispanic White population.

From this result, it appears that racial as opposed to age or gender demographic data is especially correlated with rental price movement behavior. I wonder if this may be because large numbers of the above populations are correlated with urban zip codes, and from what I understand, urban zipcodes took the hardest hits from COVID. I will keep analyzing this trend in some of my other models.

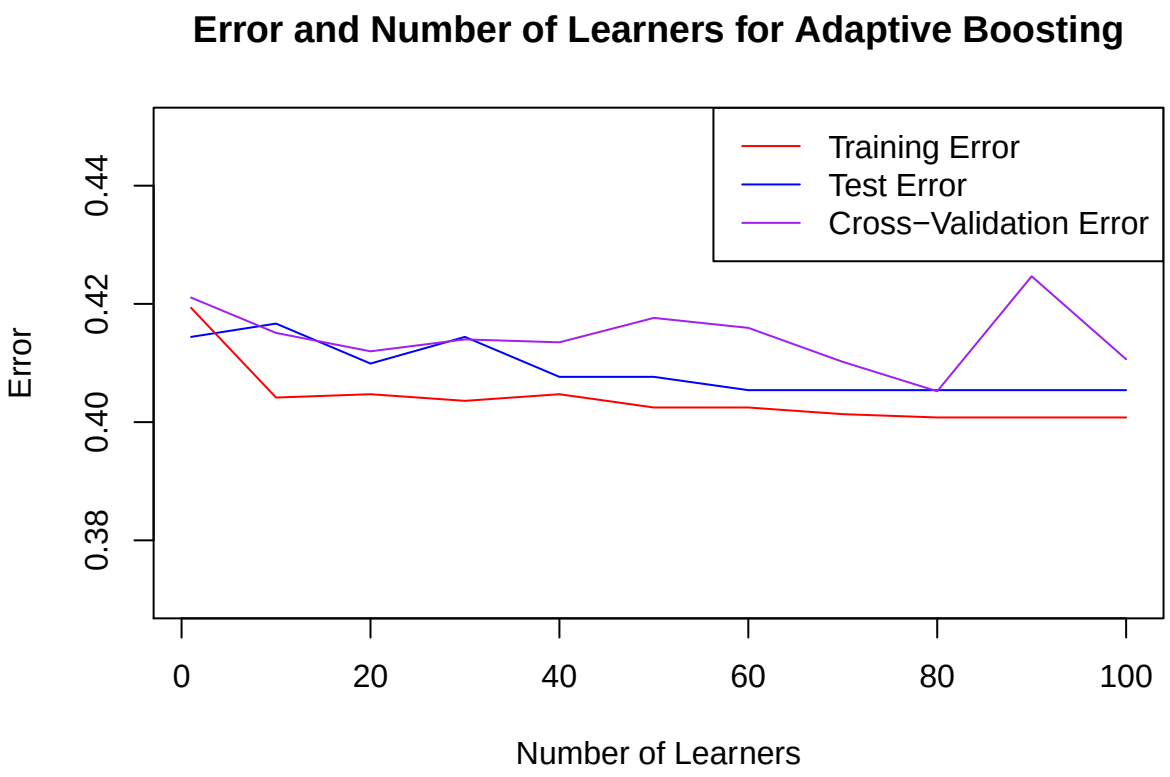
LDA

LDA seemed to perform comparably to my Logistic Regression results, which makes intuitive sense given how approximate they are mathematically.

QDA

QDA seems to perform worse than LDA on the test set despite being more complex. This indicates to me that my QDA model might be overfit.

Adaptive Boosting



From the results, it looks like my hypothesis was wrong: Adaptive Boosting performs similarly poorly to the QDA model, even with the addition of more learners.

The first decision stump seems to center around ACS question 33, which is the total number of African Americans in a certain zipcode. This reiterates what was observed in my logistic model, where race seems to be higher correlated with the outcome variable of interest than is other demographic traits like age or sex.

Results

Model	Train Error	Test Error
5-KNN	-	0.4369
10-KNN	-	0.4369
50-KNN	-	0.383
100-KNN	-	0.383
200-KNN	-	0.383
Logistic	0.332	0.376
LDA	0.340	0.383
QDA	0.305	0.432
1-ADA	0.421	0.408
10-ADA	0.405	0.406
100-ADA	0.403	0.396

From the summary of errors for the different models, I would choose to use the **Logistic** or **LDA** models.

Why these?

The LDA and Logistic Regression models have comparatively little complexity while even performing better on the test data than some of their more complex peers. This makes them more ideal solutions according to the bias-variance trade-off. Logistic is also easily interpretable on the basis of individual features.

Why not the others?

The KNN model seems to improve with higher values of K, even achieving identical results to the Logistic Regression for K larger than 50. However, especially for values of K much larger than 50, I'd be scared of underfitting. Results from KNN regression model summaries are also generally less interpretable.

The QDA model seems to be overfitted given its comparatively low training error and comparatively high test error.

The Adaptive Boosting Models see only minimal returns to additional learners, and adding ones beyond the current maximum of 100 seems like it could lead to overfitting.

Conclusion

All in all, I'm not particularly proud of my models' performance. The variations in performance across the 11 models that I studied was not particularly significant.

Furthermore, consider the most naive of predictors, a simple Bayes classifier best off the population proportions. Applied to the test set I chose, this Bayes classifier would have a 0.39 test error simply by assigning every zipcode to have decreased rent. That my best performing model only beat this naive classifier by 0.05 is a bit humiliating.

This leads me to make the following conclusions:

1. In the future, I should try and perform more in-depth preprocessing and feature engineering. Perhaps the classification models I created were being misled by nonsignificant features. What this doesn't explain, however, is why my adaptive boosting algorithm failed to produce better results. Adaptive boosting should've only focused on the significant features, and appropriate cross validation of the ideal number of learners should've removed decision stumps using misleading features. However it's possible that it wasn't so much a byproduct of the inclusion of the features but attributes of the features themselves. Perhaps I could've benefitted more from more in-depth outlier analysis or exploring more in-depth the correlations between the seasonality I observed and the outcome variable. I also could try and examine how my choice of normalizing the data may have damaged the predictive capabilities of my models.
2. It's possible that the ACS data simply isn't a good predictor of the outcome variable. There may be a reason that no Machine Learning Repositories use the above data as an example of a viable classification problem.

The fact that the Naive Bayes classifier acheived better performance than Adaptive Boosting might be a sign that there isn't a particularly significant correlation between the observed demographic data.

All in all, I feel that there is still more to be done with this dataset. I feel that further approaches to the work that I've done so far would benefit from observing some of my mistakes in this analysis, so it was not all for naught.