**Presentation Best Practices**  
A small slide deck may help summarize key points. Use it in conjunction with your R Notebook - in the presentation switch between them as needed. Be sure to pause recording when switching, setting up, or when long computations are performed.

* **(0:30) your business problems, i.e., what you are trying to predict**

For my project, I’ve decided to tackle a complicated urban issue – gentrification. Specifically, can statistical forecasting models be used to predict which urban census tracts are likely to see gentrification in the near future?

Gentrification is the process of higher income households moving into lower income neighborhoods. While these higher income households can bring investment and economic growth, they can also drive up housing costs and cost of living in the neighborhood, and this can price lower income households out of the area. For policymakers wishing to ward off or limit this displacement of lower income households, it might be helpful to have a way to predict which areas are likely to see this kind of pressure.

Luckily for me, there is a lot of research about what drives gentrification, and how to measure it once it happens. More details on these studies are in my RMarkdown document if you’re interested, but a few general summary points are helpful. First, most of these researchers focus on highly localized data, and less on data that frames the regional context that the neighborhood sits in. Second, most of this research finds that gentrification is incredibly difficult to predict until it’s begun to happen. The first point I can address in my research, the second will make me feel better when my models also fail to have strong predictive power.

* **(0:30) overview of the data and where you obtained it**

My data set is compiled from a number of sources, mostly from the U.S. Census Bureau. One incredibly helpful resource I located was from Brown University, which provides census tract level demographic and housing data across all decennial census years, and they do the hard work of standardizing all of the geographies to 2010 census tracts. This makes intertemporal analysis much easier. I downloaded this demographic and housing information from the 1990, 2000 and 2010 census from them. The 2000 data set will be used for model development and the 2010 information will use those models to generate a prediction for tracts that are likely to see gentrification by 2020.

My second source was directly from the U.S. Census Bureau, where I downloaded census information on the metropolitan level, as well as a geographic crosswalk that allowed me to join the metro area to the census tracts that comprise it. That way I can include these metro level statistics in the census tract level data.

Third, I wanted my model to include data points on transit accessibility and general mobility and connectedness, hypothesizing that locations that provide residents with the best or most broad access to as many locations as possible would be the most likely to gentrify. In order to do this, I downloaded census-tract level information for 2010 from the [EPA’s Smart Location Database](https://edg.epa.gov/data/PUBLIC/OP/SLD). This data set includes data points such as walking distance to a transit stop, number of locations accessible by transit, and number of locations accessible by car. Since these data points are on the tract level, joining them to my data set is straight-forward.

The most important data point I needed was an indicator for whether or not a census tract gentrified between 2000 and 2010 so that I can have an output variable for my testing data set. No direct data download was available, but I adopted a variation of the approach that Governing.com took in their [gentrification analysis](http://www.governing.com/gov-data/gentrification-in-cities-governing-report.html) to calculate for myself which tracts gentrified during the time period.

* **(1:00) how you explored the data**

Data was explored for two reasons: technical joining of the data and relationships between data points

Data relationships explored mostly through correlation plots (via pairs.panels) and basic charts that sought to provide information about relationships between variables.

Data exploration was also heavily focused on feature selection. By studying the pairs panels and scatterplots, I was able to reduce the number of features in my data to the ones that seemed most impactful or potentially interesting.

<show a few examples of the pairs panels and other plots.

I did attempt a principal component analysis, which ran fine, but I did not think the principal components were strong enough to replace the individual features. It took around 20 principal components (based on 22 features) to get to about 90% of the variance. I prefer to have 22 straightforward features than 20 complex, difficult to understand features. The results of this Prinicipal Component Analysis were an initial sign that perhaps the features in my dataset might not have very strong predictive power.

* **(1:30) what kind of transformation you needed to do - briefly show code**

There were a lot of transformations and adjustments that needed to be made in order to join and normalize the data. I won’t share all of them, but they are all documented within my code if you’re interested. A few of the more significant transformations included turning all of the totals from the census tract file into proportions of the whole. Since I am modeling census tracts of different sizes and densities, measuring variables as proportions of the whole is essential to get an apples to apples comparison within the model.

A lot of work also went into the geographic crosswalks between metro areas, census tracts, and census tract block groups. All of these have complicated one-to-many relationships, and then these geographic definitions change over time. I spent a lot of the data preparation phase ensuring that my primary keys were perfect for the joins.

Beyond that, there was outlier detection, data imputation of missing census tract level information (based on the average from the rest of the gentrification-eligible tracts within the same metro, and derived feature creation. One of the more interesting features that was derived is a measurement of whether a metro area has added as many housing units as households over the prior decade. The thinking here was that if a metro gained more households than housing units over the prior decade, then it would put additional pressure on the housing market, which could increase gentrification pressures.

* **(1:00) what models you built and why**

Because my outcome variable is a binary classification, but all my predictor features are continuous numeric, there are a few models that are obvious choices. For my models, I am using a logit regression, a support vector machine, a neural network, and a random forest.

I chose these models because I wanted to choose models that learn differently, and set different types of rules and decisions for sorting the data. The hope was that these different methods would have different strengths in classification, so that when they were combined into an ensemble model they would be able to make up for each other’s shortcomings.

When I could, I used k-fold cross-validation when training the models before running the validation data through the models.

* **(1:30) how the models performed -briefly show running code**
* **(1:30) how you evaluated, validated the models**

Starting with the logit model. I started with all of my features. The output was pretty promising, with many of the features showing a statistically significant relationship, including features from each of my starting data sets, like transit accessibility, metro housing production pace, and a number of the census tract level housing and demographic characteristics.

But there were some features that did not have statistical significance, so I decided to backfit the model using a step function, which resulted in a simpler model with similar impact. I then trained this model with 10-fold cross validation. Running the validation data through the model, we see it achieves \_\_ accuracy, but when you look at the true positive rate, it’s very low, at \_\_. This shows that the model is not particularly good at identifying the tracts where gentrification took place. I am using Area Under the Curve to evaluate all my models, which lends itself nicely to a binary classifier. For this logit model, the AUC was not much better than 0.50 or random chance.

Next up was the support vector machine model. I ran this using the ksvm() package, and tried pretty much all of the different kernel options, with the packages built in cross validation. In the end, after running the test data through all of the kernel options, the rbf model had the best overall accuracy and AUC performance. However, the performance, again, was not much better than 0.50 AUC.

Onto the neural network, which I won’t run here since it takes a lot of time. I ran the model several times with different numbers of hidden layers. I ended up going with 5, which resulted in another model with relatively weak accuracy and AUC.

Random forest is the final model I ran, and I ran it with 500 trees. This model performed a bit better than the others at first glance, but still not terribly strong.

* **(1:30) how you built an ensemble model and how well your ensemble performed**

So, onto the ensemble learner. I tried this two ways. One was a simple averaging of the outputs. So, I created a dataframe from the predicted values of the various models, created an average column, and checked that average column against the test data. There was a slight uptick in accuracy and the AUC measure, which managed to break 0.60.

The second way I tried this was to create a stacked ensemble, using the underlying model predictions as independent variables in a logit model that would predict the outcome. Because this second layer logit model was trained using the validation data, I then tested it using a third partition of my data that I had set aside just for this purpose. I ran this test data through the first layer of models, and then ran those predictions through the second layer. Although this was kind of a fun and interesting new experiment for me, unfortunately the ultimate result had lower accuracy than the averaging ensemble.

* **(1:00) summary and key lessons learned**

The biggest lesson I learned is that gentrification is an incredibly hard phenomenon to predict with readily available data sources. This, of course, is not a surprise. However, this is a bit vexing because people with knowledge of a city can often guess which neighborhoods are changing and gentrifying. There are certainly other indicators that other researchers have thought of, including movement of a “creative class” into a neighborhood, or an increase in the number of certain businesses, such as dog groomers or coffee shops. Data points like these are harder to find, especially for a modeling process that includes locations around the country and over multiple periods of time.

Although my model was not particularly predictive, I still wanted to take the next decade’s data (that’s 2010 data) and make a prediction for which census tracts would gentrify in 2010. So, I ran the same data prep and selection processes as I did with the 2000 data, ran the data through the first layer models and then my averaging ensemble, and spit out a set of predictions. The model suggests that \_\_\_ census tracts across the large metros are at risk of gentrification by 2020.

If my model had been more predictive, this would be the fun part. Being able to map and visualize where those census tracts that are deemed “at risk” might have been informative. Still, I wanted to demonstrate how this kind of deployment would work, so I’ve built an app in Shiny that allows for a metro-by-metro visualization of which tracts are predicted to gentrify by 2020. I’ll leave the link with the other materials.

Thanks for your time and I appreciate any feedback.