Annie Adams, Anum Damani, Andrew Bissell, Eric Cha

Professor Singh

PSTAT 197A

6 December 2021

Project Part 4: Summary

For our project, we attempted to predict case numbers for different counties in California. We extracted one years worth of data from five different signals within the Delphi Epidata API - March 2nd, 2020 to March 1st, 2021. When we performed the time series split of the data, we used an 80% training and 20% testing model so our testing dates were approximately from December 31st, 2020 until March 2,2021. While the actual COVID case count in each CA county was our ground truth, we used data from the Change Healthcare and Doctor Visits sources for our features.

Starting with the signals we pulled from the Change Healthcare source, *smoothed-outpatient-covid* displays the percentage of doctor visits with confirmed COVID-19 and *smoothed-outpatient-adjusted-covid* is the same but with systematic day of week effects removed. Something to point out about these two sources is that, because they are so similar, approximately 4,000 of the values from both are the same. We also used *smoothed-outpatient-cli* which reports percentages of outpatient doctor visits with COVID-related symptoms, but not necessarily confirmed COVID. Our other two signals were from the Doctor Visits source, *smoothed-cli* and *smoothed-cli-adjusted*. These display percentages of outpatient doctor visits where patients came with COVID-related symptoms, the latter of which has systematic day of week effects removed. Overall, all the data we pulled from these signals is similar in that they deal with visits to the doctor where the outpatient either had COVID-related symptoms or were a

confirmed case of COVID, which we can associate with case counts within the counties. In addition to that, we also chose these signals because they contained data for the time period we were looking for. We originally intended to make use of signals from the Google Symptoms source too but ended up phasing them out of our primary dataset for multiple reasons. One, this source only contained data for a period of 9 months which would force us to impute data for 3 whole months. While this is a feasible issue we could resolve, we opted to pick a signal that contained enough data as opposed to figuring out a way to impute all of the missing data, with the hopes that it would increase our performance. The other issue with Google Symptoms as well was that there was a lack of data for all of the counties; because only 15 counties worth of data were provided by the source, we felt that our predictions wholistically for the state of California would be lacking as a result of continuing to make use of these signals. We also realized that conceptually, symptoms like ageusia or anosmia do not directly deal with case numbers because they can be direct results of other illnesses like the common cold, and thus might not be the best data to use to make predictions on case numbers for COVID specifically. All in all, these were the 5 sources we chose to use. Once we gathered these 5 signals' value columns into one data table, we imputed all the missing values using the column averages. Then, we set the original as the t-step in our time series and then manually inserted t-1 step columns for each via excel spreadsheets. We then exported the csv file and imported it into our DeepNote workspace for this project to begin the training phase.

For our interpretable model, we chose to use a decision tree regressor with cross validation. After defining our X, y, and our time series split object, we conducted training on a Linear Regression model across 5 folds and computed the R^2 and RMSE for each fold and an average of both for the model as a whole. Initially, our results were not desirable in the slightest

due to the fact that we were including the dates as one of our features and we were using linear regression with the FIPS codes as values in the data rather than as features - this yielded skewed or negative performance results. Subsequently we pivoted towards utilizing a Decision Tree Regressor instead, with the hopes of obtaining a lower error/higher R^2. After dropping the dates as a feature and switching models, we significantly improved our error and our model achieved a R^2 of approximately .87 on our validation set and .97 on our test. Our RMSE for our validation was approximately 186 while our RMSE for our testing was approximately 197. It took us a bit to wrap our heads around how to properly interpret our results. Our two methods of understanding our scores involved looking in inverse directions, i.e. our R^2 decreased for our testing model, but our RMSE improved and vice versa for validation. For our hyperparameter tuning, we varied the depth of the decision tree to see what depth performed best. After several trials, we determined that a depth value of 3 more consistently gave us the highest R^2 value compared to the other depths, and then fit our model on this depth. As this depth is quite high, it is likely that our model overfit a bit and alludes to why our R^2 is quite high.

For our complex model, we implemented a Support Vector Machine Regression (or SVR) model with an rbf kernel. Our model originally performed pretty poorly with an R^2 of approximately .5 for our validation and .03 for our test. However, after normalizing the data, our results improved significantly. For our validation score, we got an extremely low error, but also a fairly low R^2 (.6 RMSE and .53 R^2). Our results improved for our testing, similar to as we saw in our Decision Tree Regressor - for our testing set, we got a RMSE of 913 and a R^2 of .54. These were analogous to our result when we ran the same model but with a linear kernel rather than an rbf kernel, however our R^2 was extremely high for the validation/testing R^2. This can be attributed to the fact that a radial basis function kernel is often used for classification and we

were using it for regression. Our data could also be linearly separable, thus causing the linear kernel to perform better.

At the end of the training phase, when our team felt comfortable enough with where we ended up with our results, it was time to determine what kinds of plots to come up with to display the results. For the interpretable model we started with plotting out our validation MSE and R^2 score averages across the validation set to demonstrate our hyperparameter tuning, and then proceeded to plot our predictions against the ground truth class data. We first plotted the predictions for the entire year period against the entire ground truth class, then developed a way to plot out predictions by county despite having dropped the fips codes before training began. To do this we essentially re-examined the dictionary object we created earlier in the data-preparation phase of the project containing the CA counties and their matching fips codes. This gave us the order in which our data was organized for each county, and so to plot predictions by county we only needed to manually index our predictions by selecting every 59th row of the data (adjusting for the CA state fips code at the beginning of the dataset). We did this for our Decision Tree results as well as for SVR.

Overall, our results from the standard vector regression model with a linear kernel had the best results, as they were insurmountably high. Therefore, our complex model performed better than our interpretable model. This is to be expected as more complex models are often more accurate. Just like a bias-variance tradeoff in models, there also exists an interpretable and complex tradeoff. Interpretable models are ideal because they allow us to interpret our results and understand our data better. If we were to pick a model that was interpretable yet also complex and yielded a high accuracy score, we would pick a decision tree regressor with a max depth of three. This model still performs well while also being more interpretable that our

standard vector regression model. It is also worth mentioning that the decision tree model responds better to outliers than a SVR model with an rbf kernel, demonstrated by the more accurate predictions of case count spikes in our plots.