FORECASTING COVID CASES USING COVIDCAST

Alyssa Keehan, Jasmine Kwok, Jordan Tran, Tiffany Hsu
University of California, Santa Barbara
Data Science Capstone

Abstract / Introduction

- This project aims to forecast future COVID-19 cases by building predictive models given past data.
- COVIDCast Epidata API provides county level data on the spread and impacts of the pandemic across the United States.
- We use available data for doctor visits, trends and surveys, google searches, hospital admissions, positive antigen tests, time spent at home, and COVID cases to train the models.
- We found that with our final linear regression model, no single feature had a larger weight on the result more than any other. (i.e. all coefficients were relatively the same) but all of them were significant to the predictive model (Pr(t) < 0.05).
- A LSTM model performed best over other neural network architectures. Model struggled with overfit and much regularization was needed to produce a lower RMSE.
- This project is motivated by the lack of previous knowledge surrounding the understanding of COVID spread, which led to mutations and prolonging of the pandemic period.

Research Questions:

1. Given the past data, can we build predictive models that forecast the future of the pandemic so that we can see one step ahead and prepare accordingly?
2. Which data is highly relevant to the prediction and how should that affect our policies?

Data Preparation/Cleaning

Selected Features (t, t - 1):		
Geo-values (county)	COVID visit	Google symptoms
COVID cases	Doctor visit	COVID deaths
Confirmed doctor visits	Hospital admissions	Home time

Response Variable (t + 1): Total COVID cases

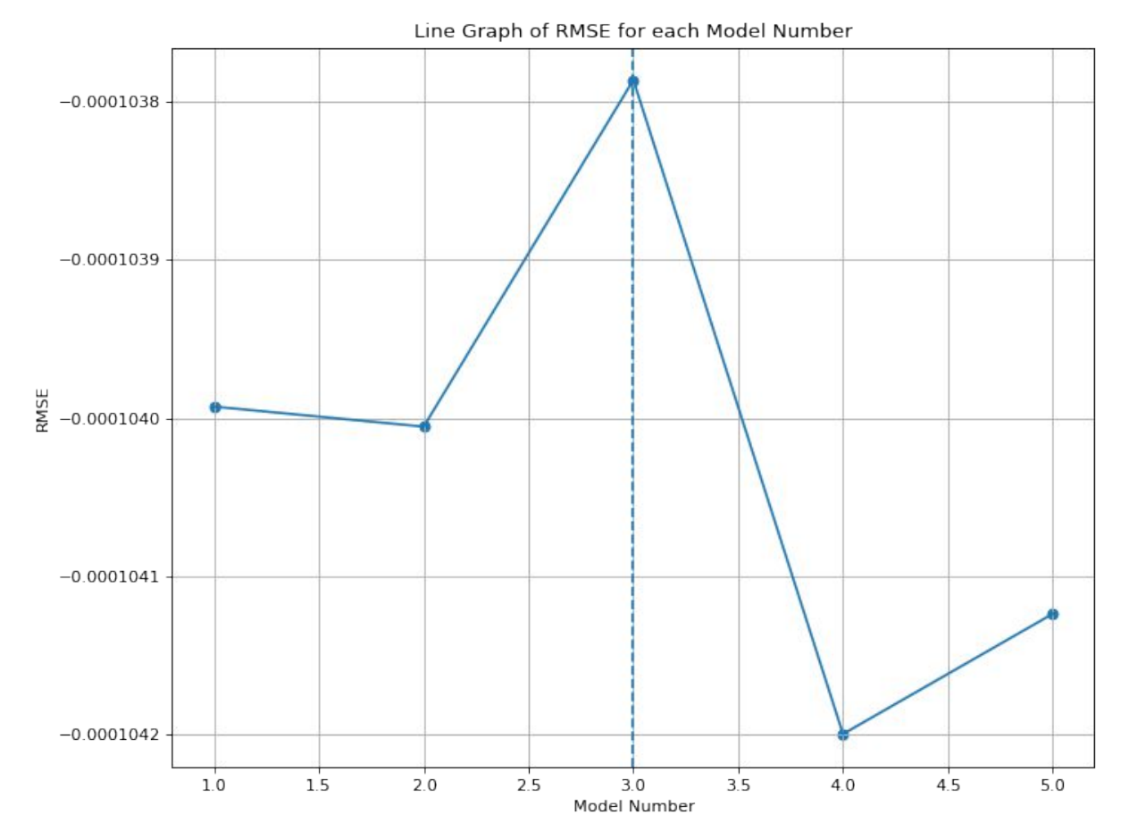
We obtained our data from COVIDCast Epidata API which provides COVID-related data from various sources.

1. Generated a dataframe for each feature with data from February 1st, 2020 to November 1st, 2021
2. Merged the data frames according to geo-values and dates
3. Imputed the time series by forward filling with a limit of 3 days
4. Created multiple datasets with different features and amount of timesteps

Model: Linear Regression

Method:

- Backwards Subset Selection on the imputed full data
- Cross validation to compute the rmse for each proposed and subsetted model



OLS Regression Results

	coef
geo_value	-4.027e+09
covid_cases	1.6091
doctor_visits	0.0030
hospital_admissions	0.0024
covid_deaths	0.0396
home_time	-0.0002
geo_value_t_minus_1	4.027e+09
covid_cases_t_minus_1	-0.6077
confirmed_covid_vist_t_minus_1	-0.0008
doctor_visits_t_minus_1	-0.0019
hospital_admissions_t_minus_1	-0.0024
google_ageusia_volume_t_minus_1	0.0010
covid_deaths_t_minus	-0.0412
home_time_t_minus_1	0.0004

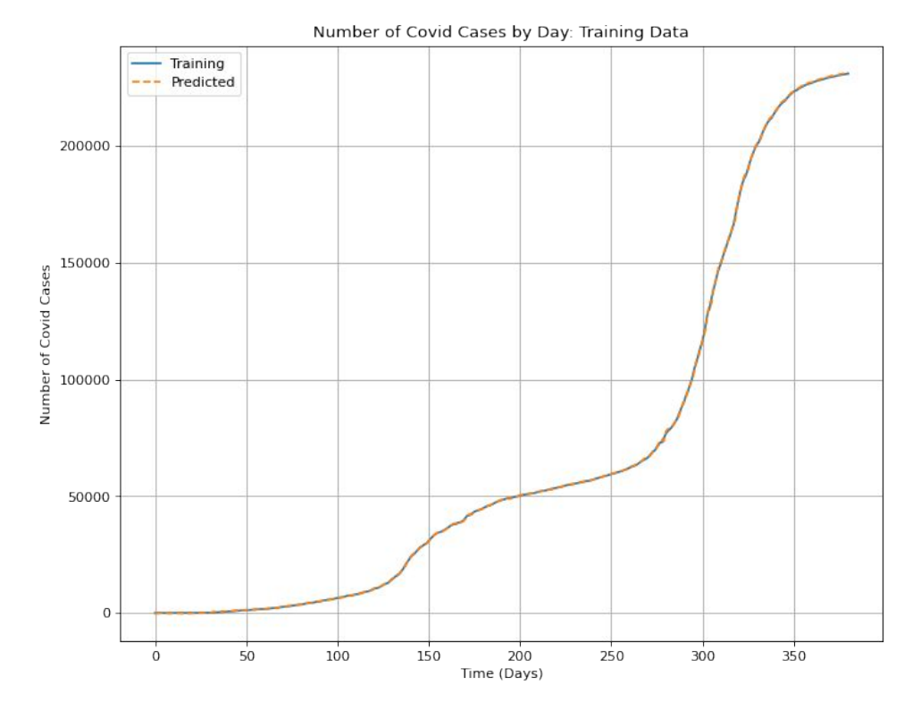
Findings:

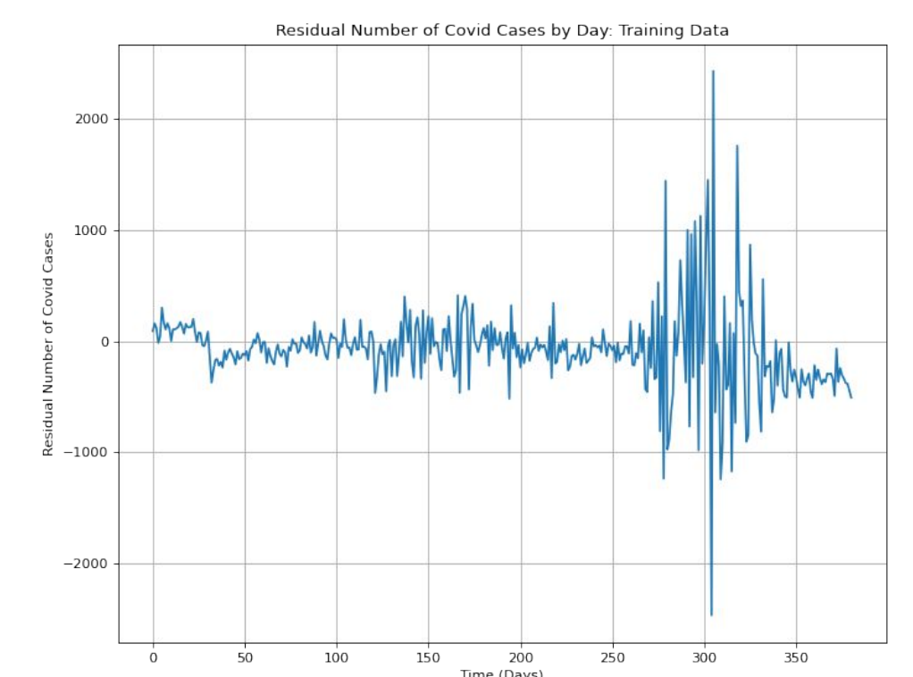
- After performing the subset selection 5 times, we noticed a minimum rmse in the 3rd model, so we will continue to apply and test using that model
- The RMSEs for each of these proposed models were all very small
- The features included in the final model are listed on the left

Results:

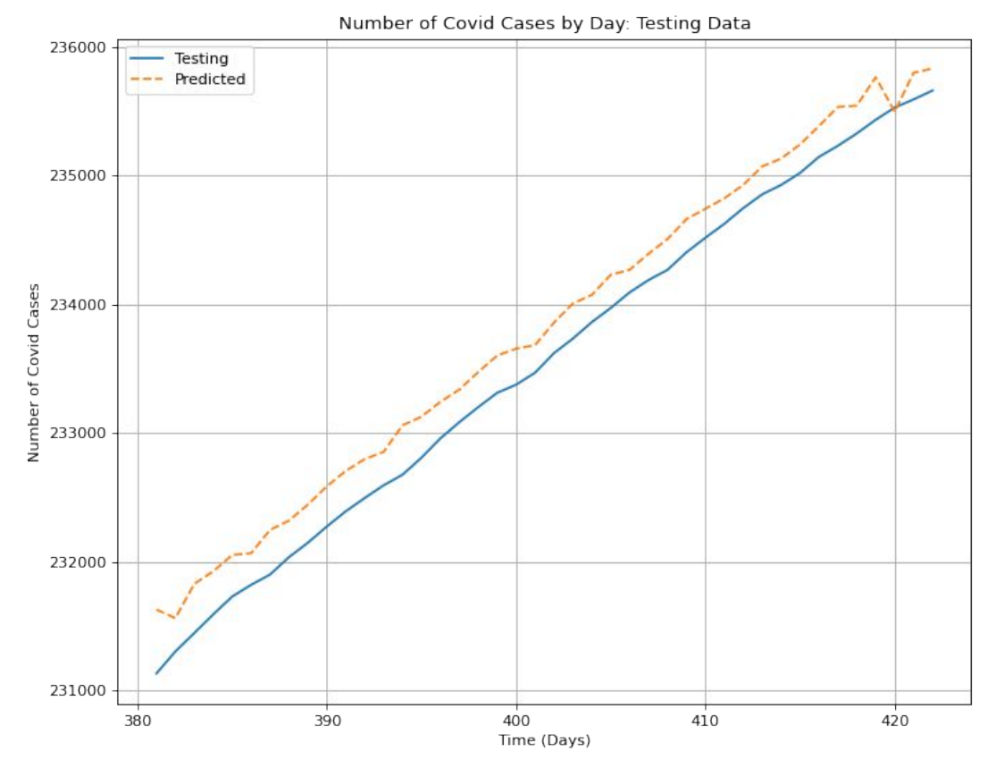
```
Train RMSE:  857.6800823455156        Train $R^{2}$: 0.9999613577220372
Test RMSE:  356.292264790975          Test $R^{2}$: 0.9999982766904577
```

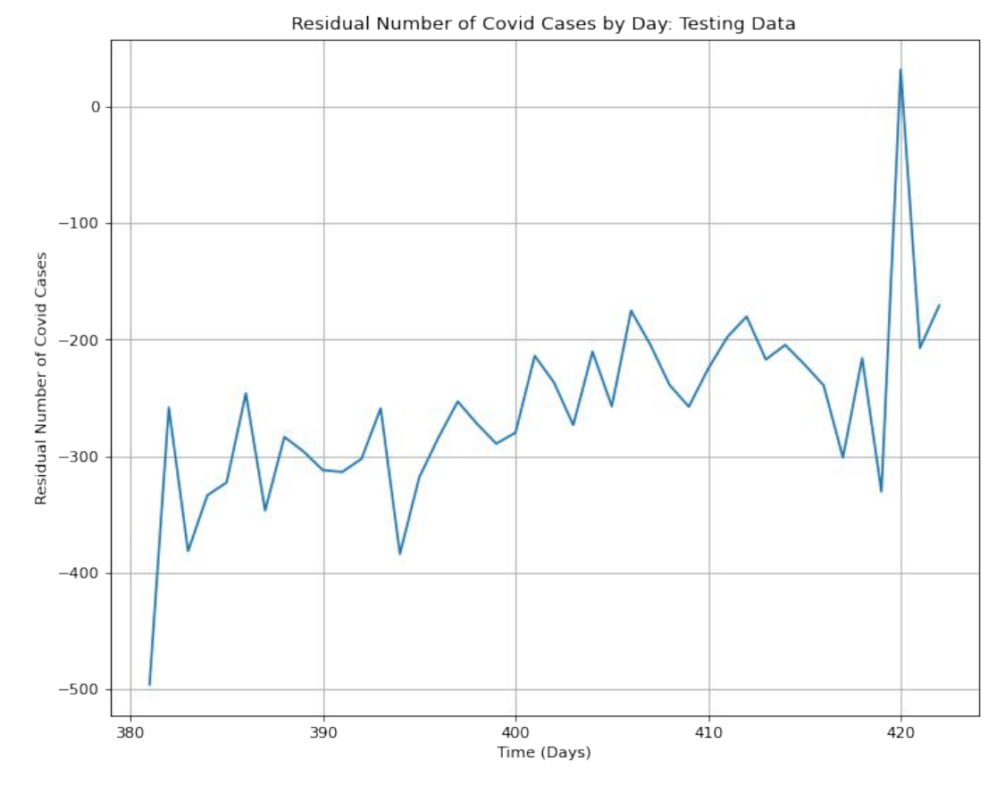
The plots below show the training, testing and combined data for Orange County COVID cases as well as their corresponding residual plots.



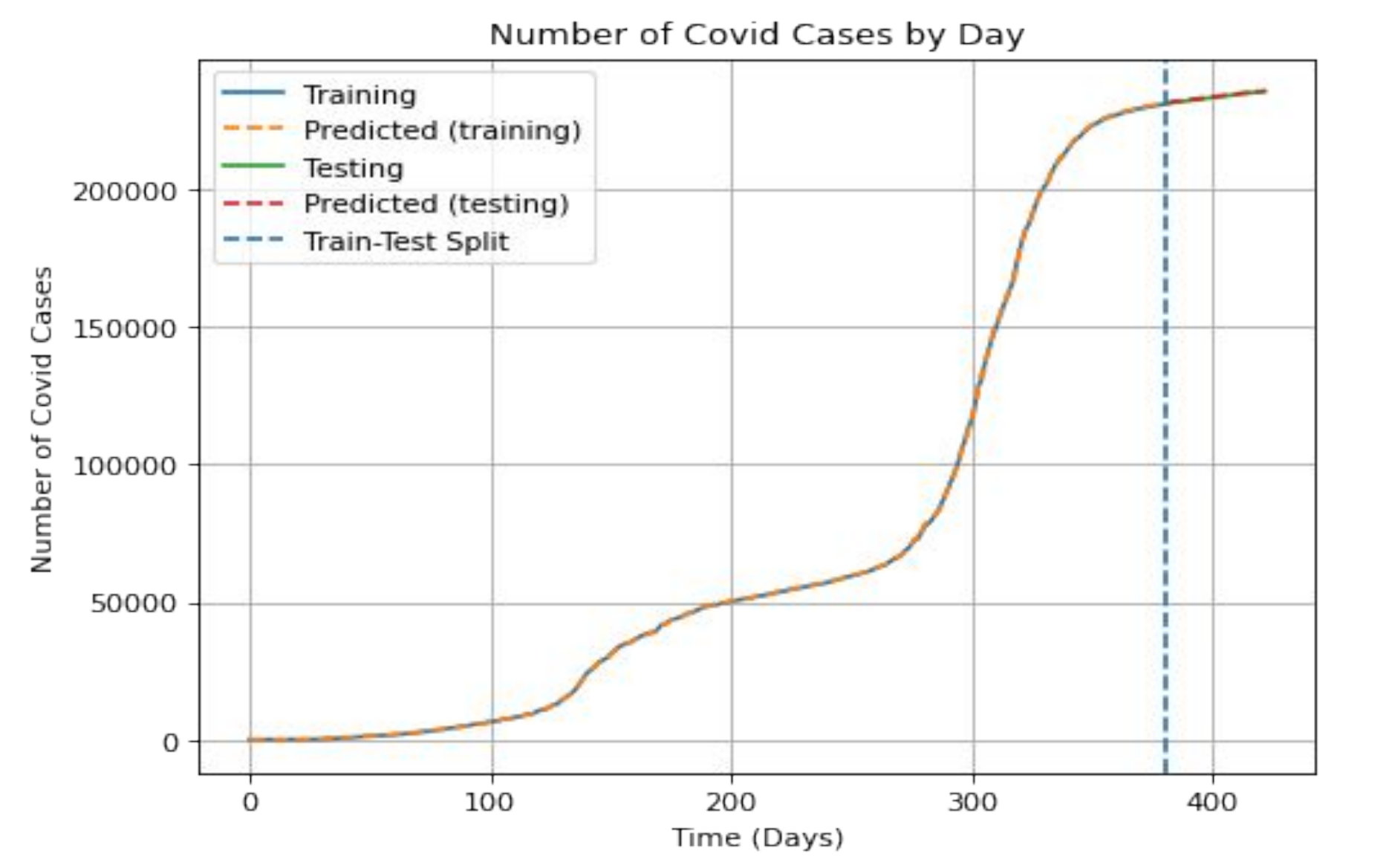


Since the model is trained on the model, we expect the predictions to be very accurate and close to the true predictions.





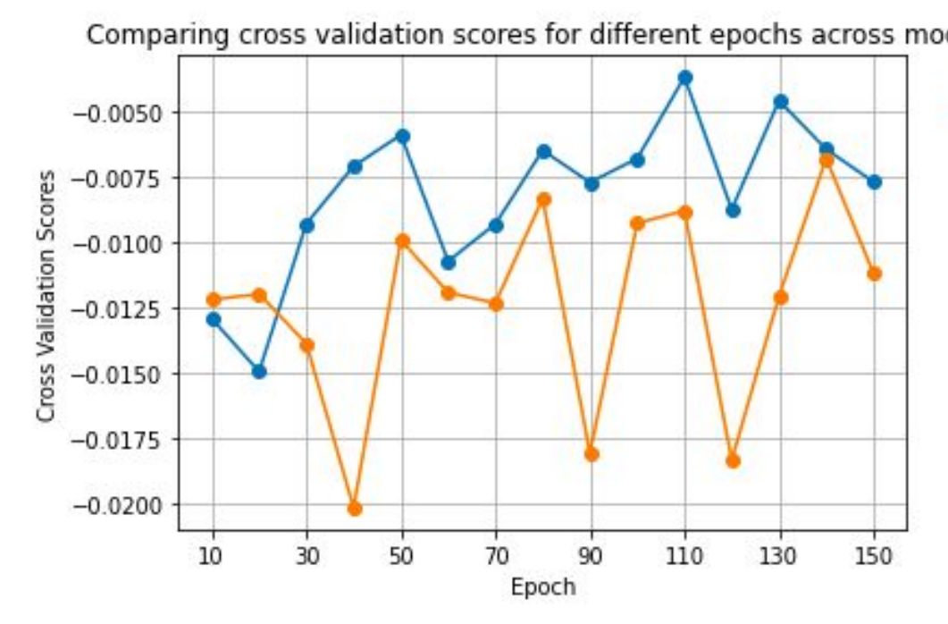
Since the testing data is unseen to the model, we expect there to be some differences between the true and the predicted values. It looks like the model is underestimating the number of COVID cases.

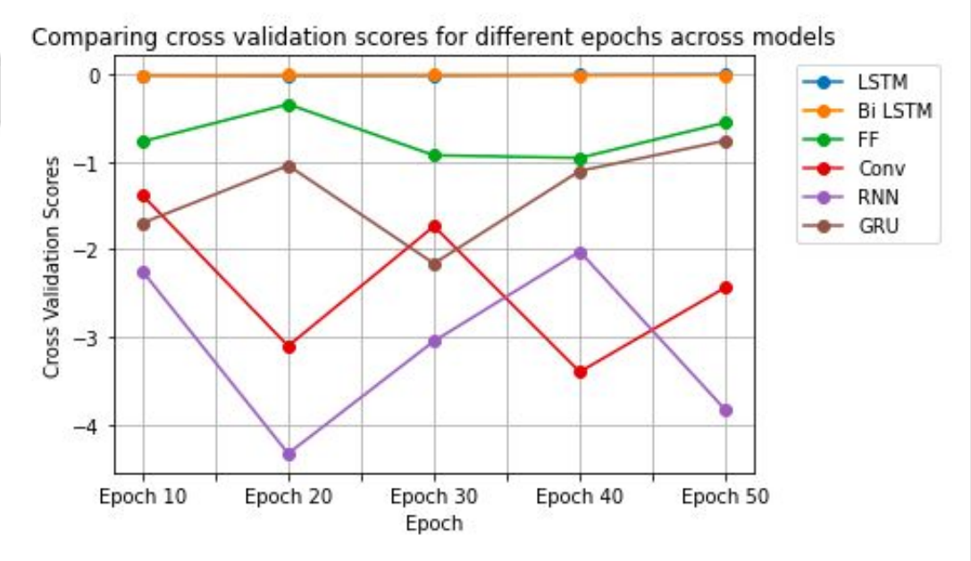


Model: Neural Network

Method:

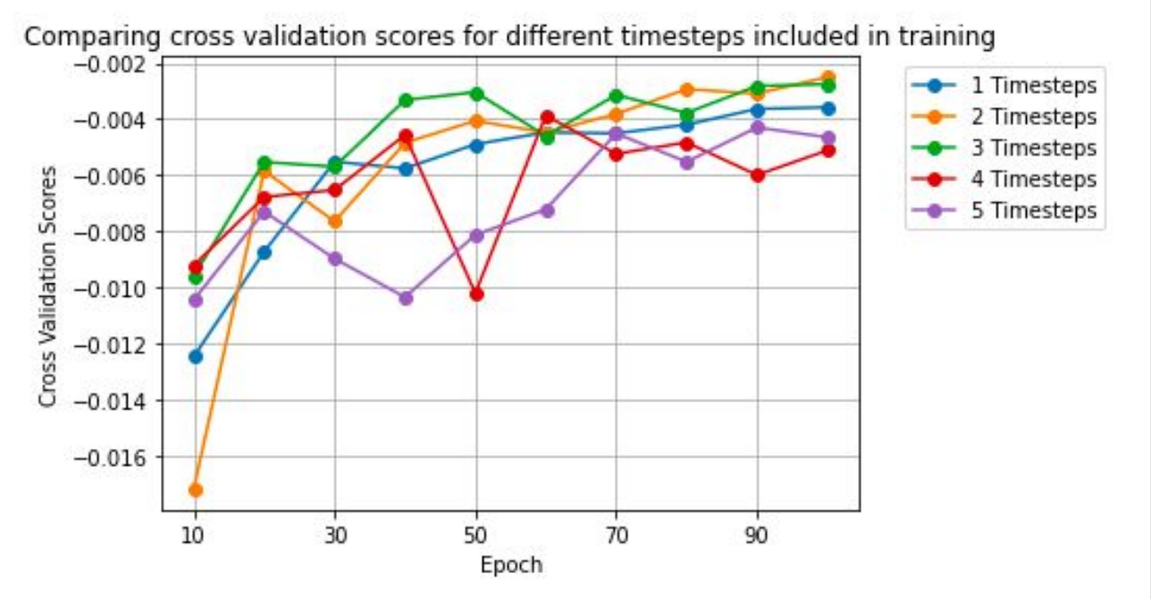
- Cross validation to choose the best model over different epochs
- Cross validation to choose the best number of time steps and features
- Hyper-parameter tuning via randomized search cv grid
- Test root mean squared error as accuracy metric
- Bagging Ensemble

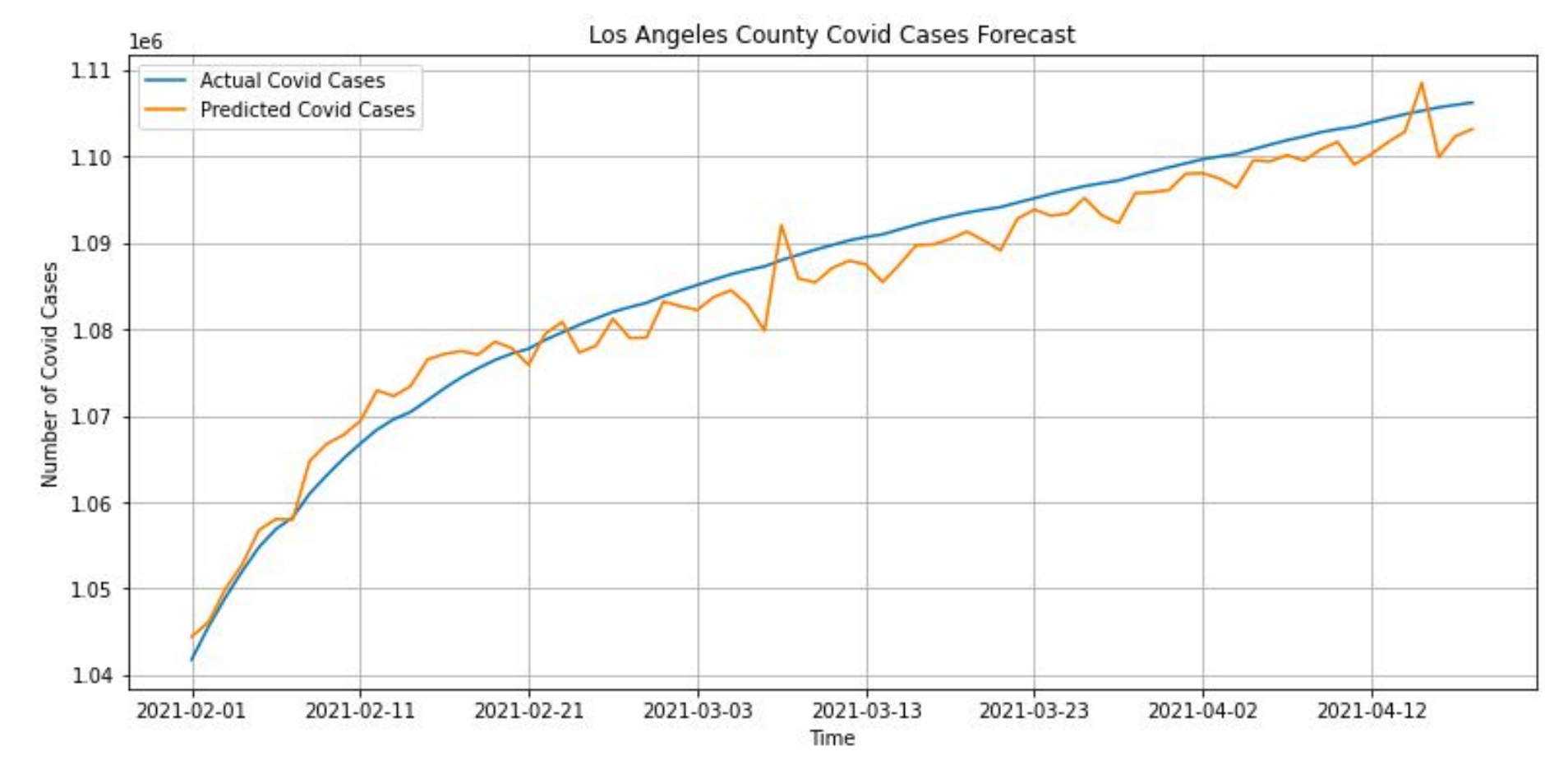




Findings:

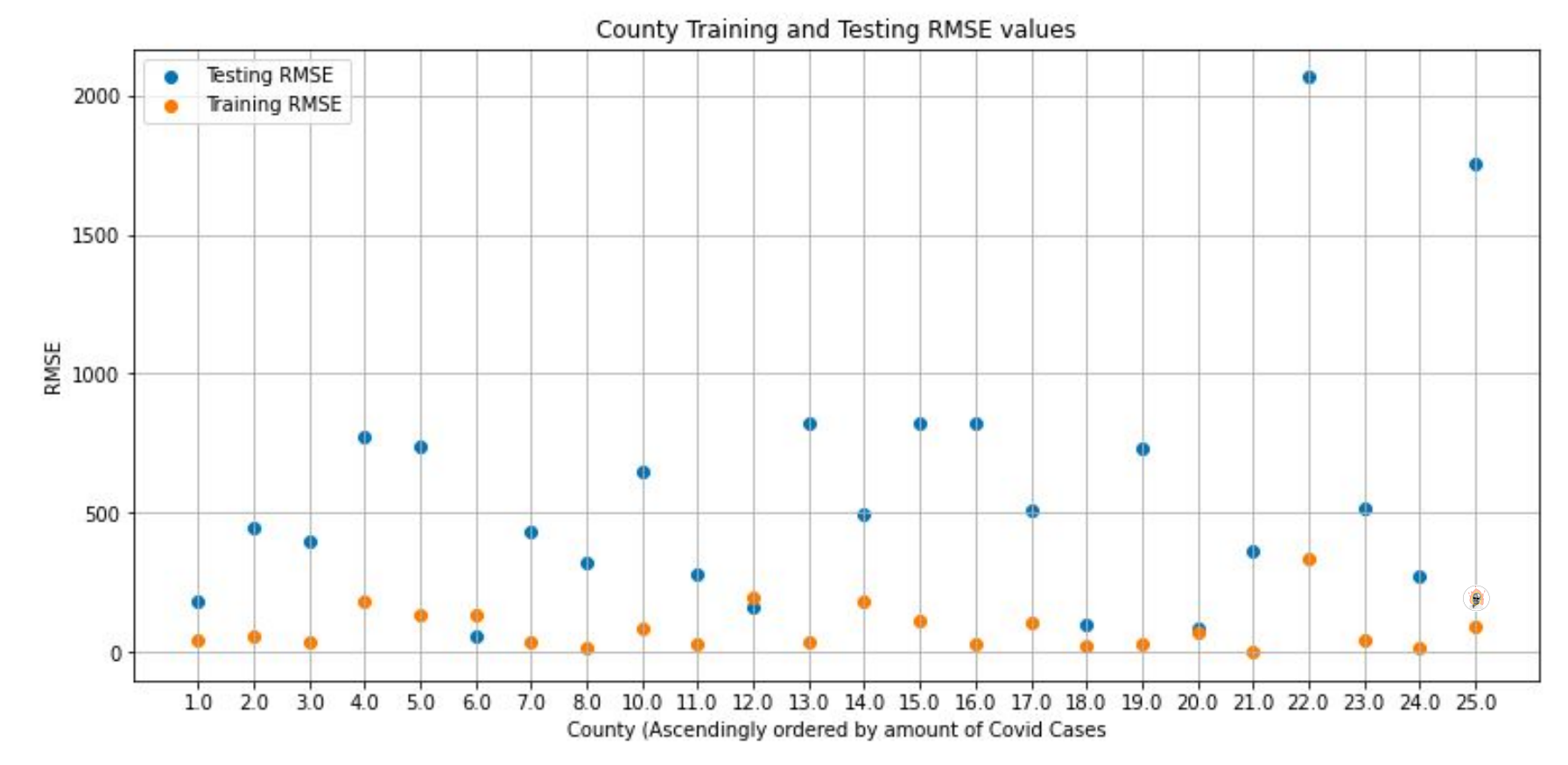
- LSTM model performs the best across all 6 models
- Google symptoms decreased prediction accuracy (removed)
- At lower epochs, time step 3 performed the best. However, at higher epochs, time step 2 outperformed time step 3.
- Best hyperparameters are a 96 unit single LSTM layer, softplus activation function, SGD optimizer, and a batch size of 2.

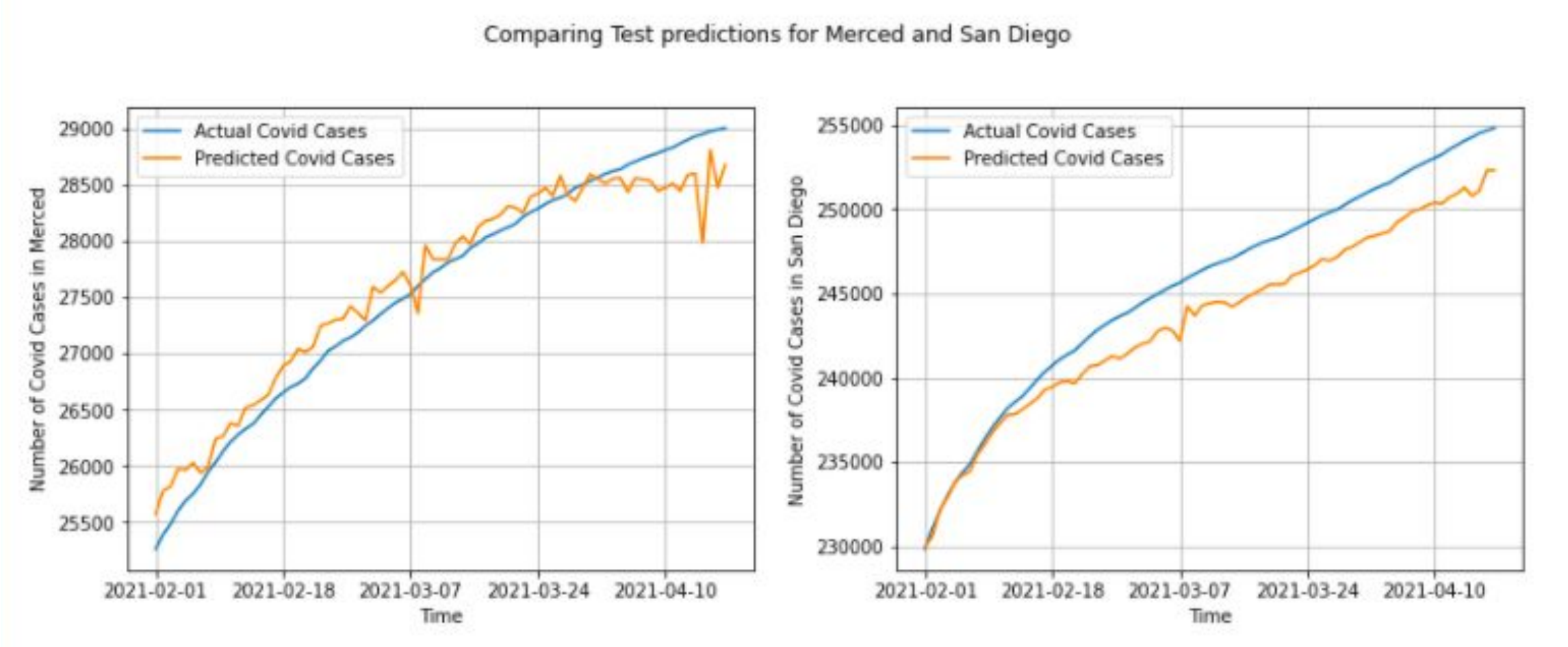




Results:

- Our final test RMSE value for the model is `RMSE: 677.7825422447407`
- Our model performs best for Merced. County and worst on San Diego county (22:6073). This may be due to our model overfitting as we can see from the graph below the testing RMSE is much higher than the training RMSE.





Conclusion/Future Work

Conclusion:

- We expected our neural network to perform better than linear regression. However, our linear regression performed better.
- We had a hard time cleaning and merging the data to include as much data as possible, especially for the neural network.
- It was easy implementing cross validation for our data since we have had practice on it in homeworks.
- It was difficult trying to figure out how to work with time series data, especially when trying to train it.
- The run time was long for neural network models.
- Comparing different models were challenging because of the various combinations of datasets, model architectures, and hyper-parameters.
- A lot of missing data for different dates and features made it challenging to produce a final dataset for our forecasting model.

Future work:

Linear Regression

- Implement Ridge Regression to reduce RMSE
- Experiment with different features and time frames

Neural Network

- Multi-time step prediction (predict t + n)
- More complex model structures
- More data, more features (e.g vaccine information, COVID variants, demographic information)

Coordination

- Jordan built and researched neural network models.
- Tiffany assisted Alyssa in researching the linear regression model and creating the poster.
- Alyssa mainly worked on researching and implementing the Linear Regression Model and how to Forecast Time Series data.
- Jasmine worked on analyzing the different models and created the neural network visualizations.

References

- https://cmu-delphi.github.io/delphi-epidata/api/covidcast.html
- https://www.analyticsvidhya.com/blog/2021/05/multiple-linear-regression-using-python-and-scikit-learn/
- https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
- https://otexts.com/fpp2/regression-intro.html
- http://karpathy.github.io/2019/04/25/recipe/
- https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f
- https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/
- https://colah.github.io/posts/2015-08-Understanding-LSTMs/