Project Proposal

Project Type

Our team will be doing the **software** option for the project.

Project Description

Our project aims to address the spread of Covid-19. The virus has been responsible for countless deaths, and the world has had to make many adjustments to accommodate its effects. If we can somehow predict or model its spread, perhaps we can begin to take preventative measures (instead of reactionary ones) and keep future damages to a minimum.

Dataset Description

Our project will be using the following existing datasets to accomplish our task:

- **COVID19 Global Forecasting (Week 1)**
  https://www.kaggle.com/c/covid19-global-forecasting-week-1
- **COVID-19 Complete Dataset (Updated every 24hrs)**
  https://www.kaggle.com/imdevskp/corona-virus-report

Proposed Approach

Our project aims to use **time series forecasting techniques** given current Covid-19 data to predict the spread of the Covid-19 virus. First we will clean and perform preprocessing on our datasets. Then, we will perform predictions using the moving average (MA), exponential smoothing, and autoregressive integrated moving average (ARIMA) methods. Specifically, we will be predicting future numbers of confirmed, deaths, and recovered cases based on numbers from prior dates.

This problem can also be cast as a supervised machine learning problem, so in our project we will also use **supervised learning techniques** to make our predictions. Specifically, we plan to use Random Forest Regression, Support Vector Machines, and a Deep Neural Network to make predictions.

Team

Deniz Ademoglu, Jeffrey Duong, Yejia Liu, Shreyans Magdum, Justin Wang

Labor Division

Deniz Ademoglu

- Markov-Chain: Even though we have some patterns in COVID exposure, some unknown times we see an increase in the number of cases unexpectedly. Using different orders of Markov-Chain, we can predict the number of cases given the situation in the last couple of days. Using the whole data, we can create a model that would predict the upcoming days COVID statistics.

Jeffrey Duong

- Random Forest: The Random Forest algorithm is useful for predicting an observation's class, and in this case, predicting the spread of the Covid-19 virus. This technique uses a large number of randomized decision trees to come together and make a prediction. The model's strength comes from having an ensemble of many different, uncorrelated predictors that cover each other's mistakes. To reduce correlation, the random forest can be built with a process called bootstrap aggregation, or bagging. Bagging takes random samples with replacement from the dataset to build the root and nodes for each decision tree that make up the random forest.

Yejia Liu

- Neural Networks: The popular neural networks architectures such as Convolutional Neural Networks are commonly used in tasks involving images or videos processing by taking features' spatial correlation into consideration. However, they have also achieved excellent performance for tabular data. In our project, I would try different architectures of neural networks with varying numbers of convolutional and fully connected layers, combining with dropout, max-pooling techniques. When it comes to tuning the hyperparameters, I would attempt the most popular fine-tuning methods including linear scheduler, adaptive gradient descent and early stopping strategies to explore the best performance of the currently used neural network architectures.

Shreyans Magdum

- Xgboost: It is an efficient implementation of gradient boosting, that is both fast and efficient and performs well on a broad range of predictive modeling tasks. XGBoost can also be used for time series analysis, provided that the time series dataset is transformed into a supervised learning problem statement. It will also require the use of a specialized technique for evaluating the model, known as walk-forward validation, as evaluating the model using the k-fold cross validation technique would lead to optimistically biased results.

Justin Wang

- Support Vector Machines: Support Vector Machines (SVMs) can be used to solve nonlinear regression estimation problems, so they are useful for time series forecasting. They rely on mapping input data into a multidimensional feature space using a kernel function and then performing linear regression. SVMs are highly flexible by tuning the regularization term used (C), as well as the kernel function to yield the best output. For the actual implementation, different parameters will be attempted and tested to see which is best for time series forecasting.

Evaluation Plan

Our dataset comes with its own labels. To evaluate the quality of our solution, we can simply split our dataset into train and validation sets and then compare our predictions against the labeled validation sets.

For example, we can:

1. train our models with 100 days of data
2. perform predictions for the next 40 days
3. compare predictions against the actual labels for the next 40 days

We can perform a similar process for all of our models to evaluate the quality of their solutions. To evaluate the correctness of our machine learning models, we will run our dataset through imported scikit-learn libraries and compare the predictions for each respective model.