ORIE 4741
Learning with Big Messy Data

Final Report
Predict Local Epidemics of Dengue Fever

Lu Cao
lc892

Elva Gao
yg357

Kinya Wang
qw92

Dec. 5st 2021

Contents

1 Problem Specification

Dengue fever is a mosquito-borne disease that occurs in tropical and subtropical regions of the world. Due to the transmission characteristics of dengue fever, more and more scientists believe that climate factors have a complex and non-neglectable relationship with dengue fever's spread. In our project, we want to get a better understanding of the relationship between climate and dengue dynamics to improve research initiatives and resource allocation to help fight this life-threatening pandemic.

The purpose of this project is to train a model to predict the number of dengue fever cases reported each week in two regions, San Juan and Iquitos, based on environmental variables including temperature, precipitation, vegetation, and etc. Past researches have all used those two locations as representatives as well for general predictions. Since these two cities have different environmental and social conditions, by comparing the models and prediction patterns of the two cities, we might be able to generalize our models to more cities and locations, or even other diseases which have similar transmission patterns. Being able to predict the infected cases would be impactful with providing a deeper understanding of the root and causes of dengue fever spread, along with helping with research and resource allocation to prevent and fight severe pandemics.

2 Description of Data Set

We used the Dengue Fever data from Drivendata.com. The data were provided in four different files, including one file consisting of features' data and one file consisting the number of dengue cases for each row in the feature file. We will be using those two files extensively as the other two files are related to competition submission format. Before we started the data cleaning process, we looked at the two files in more detail.

The **features dataset** includes data that can be categorized into five groups:

- **City and data indicators**: which include city (categorical data) and week_start_date (in abbrev. number format).
- **NOAA's GHCN daily climate data weather station measurements**: which include five different features that correspond to the maximum and minimum value, average and also the range of temperature as well as total precipitation.
- **Persiann satellite precipitation measurements**: which include one feature which is precipitation_amt_mm, representing the total precipitation.
- **NOAA's NCEP Climate Forecast System Reanalysis measurements**: which include 10 different features that corresponds to total precipitation (unit: amt/mm) , mean dew point temperature, mean air temperature, mean relative and specific humidity, total precipitation (unit: kg/m^2), max and minimum air temperature and diurnal temperature range.
- **Satellite vegetation - Normalized difference vegetation index (NDVI) - NOAA's CDR Normalized Difference Vegetation Index (0.5x0.5 degree scale) measurements**: which include four features (ndvi_ne, ndvi_nw, ndvi_se, ndvi_sw) that individually corresponds to the southeast, southwest, northeast and northwest of city centroid. All of the features that are mentioned above, unless stated, are in numerical form.

The **label dataset** includes four columns which include:

- city - categorical data representing different cities
- year - an integer between 1990 and 2010; it denotes the year that the data is recorded
- weekofyear - an integer ranging from 1 to 53; it denotes the number of week in the year that the data is been recorded
- total_cases - an integer denoting the number of cases

We noticed that our dataset includes information about two different cities, San Juan and Iquitos(simplified as sj and iq in the following analysis). We decided to compare the similarity and difference of data distribution of the two cities to decide whether we could use one general model to make a prediction.

3 Modeling Approach

We tried to tackle this problem by applying methods that we have learned in class. Our first step was to clean the data and make sure that all features can be used to compare in the same scaling thus it won't cause bias to the later running models. After data cleaning, we started with basic exploratory data analysis to understand each feature, their trends over the time and how they are related to each other. After examining their trends and relationships, we ran a time series model to see if the cases have seasonality involved. While running the time series analysis, we realized that the number of infection cases weren't solely dependent on the time frame, thus we decided to move on to other models to better understand and predict the data. In order to avoid repetitive features between week_start_date and Year+Week, we will drop week_start_date for future model fitting and analysis.We then tried two different regression models to help with predictions. We decided to use OLS model and Huber regression since one is a very common model to use for estimating coefficients of linear regression equations and the other is less sensitive to outliers in data. However, due to the poor performance of our regression models, we suspected that there might not exist a linear relationship. In order to capture the nonlinearity, we decided to implement random forest. As expected, this model brought out accurate predictions which were also proved by the results of using gradient boosting.

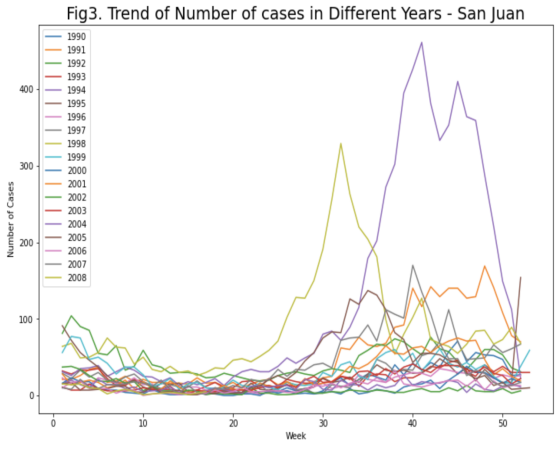
4 Analysis

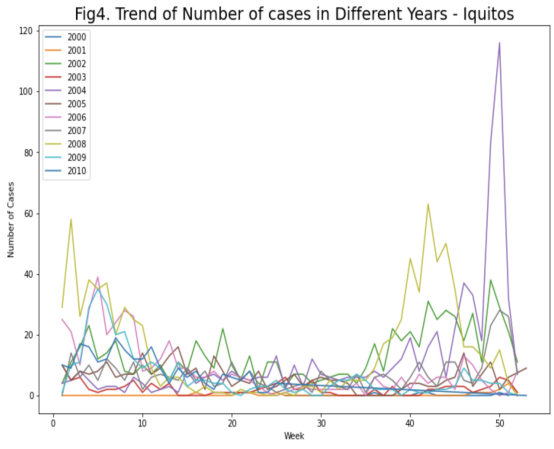
4.1 Pre-Analysis

First, we started with data cleaning: All variables were casted to appropriate types: categorical features like the name of the city were one-hot encoded. We also imputed missing values based on the average of the previous and next values.

4.2 EDA

We started the analysis with a look over the trends of different features. First we started with checking if the disease has seasonality. These two line graphs show the trends of disease outbreak in sj and iq.





Both of the graphs shown above indicate that there exists some seasonality in the number of cases increased per week as cases tend to be higher in the first and last few weeks of the year in comparison to the middle of the year. However, the trend varies between the two cities.

San Juan has a larger number of cases on average, with with relatively small Dengue Fever outbreak happens around the first 10 weeks of the year, more severe outbreak happens around the 35-50 weeks of the year Iquitos has a smaller number of cases on average, the same level of Dengue Fever outbreak happens during the first 10 weeks and last 15 weeks of the year.
The two graphs below on the next page reveal how the distribution of the number of cases each year changed over time. We can observe that the number of cases tends to decrease over time in San Juan while showing an increasing trend in Iquitos. At the same time, the number of cases increased each month tends to spread out more and has more skewness in San Juan in comparison to Iquitos. The above observation indicates that there are different patterns of Dengue Fever in San Juan and Iquitos. Thus, we decided to train a model for each of the two cities.

Fig5. Boxplot of Disease Outbreak over Years - San Juan

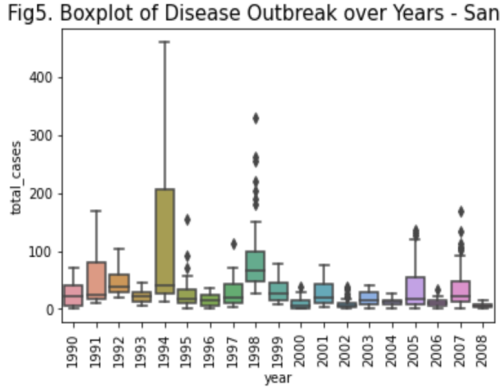
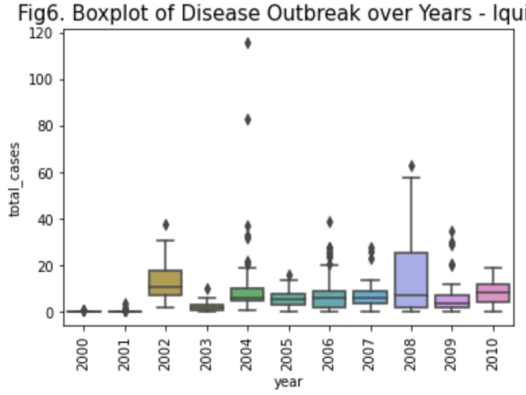


Fig6. Boxplot of Disease Outbreak over Years - Iquitos



With the goal of understanding what might be a determining factor of causing the local Dengue Fever, we created a heatmap to further examine the correlation between different features.

Fig7. Boxplot of Outbreak over Years - San Juan

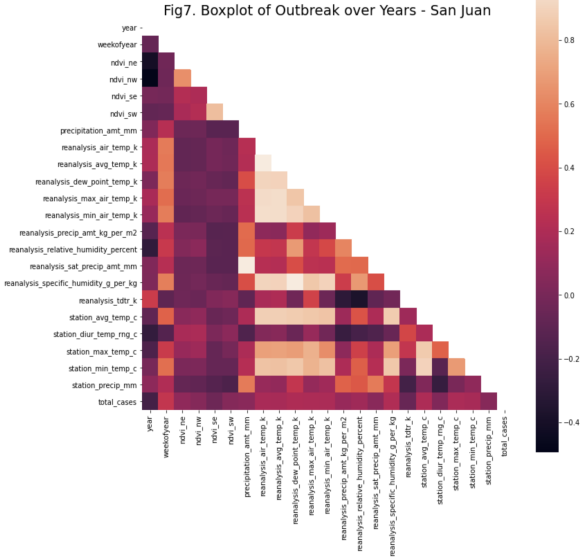
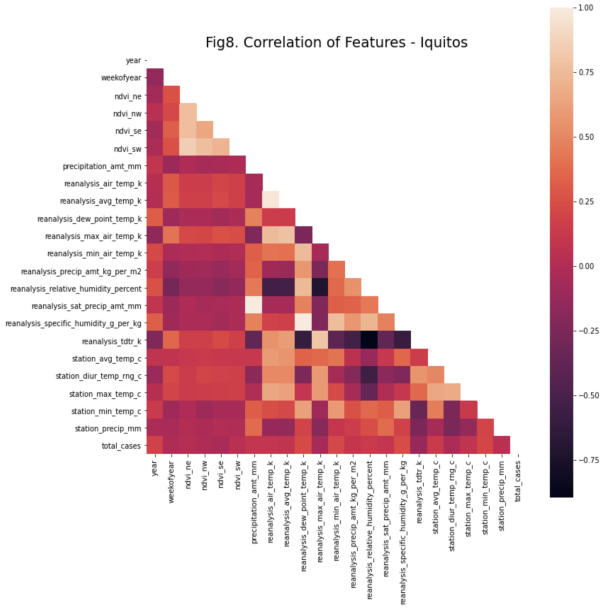


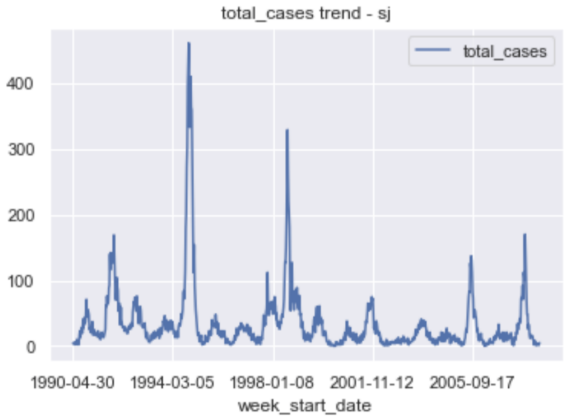
Fig8. Correlation of Features - Iquitos

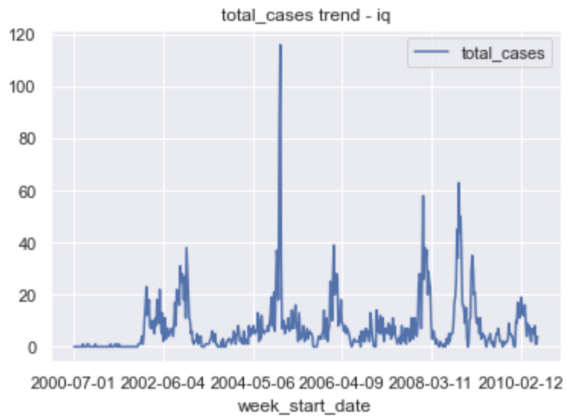


Observation: Satellite vegetation indices (ndvi_ne, ndvi_nw, ndvi_se, ndvi_sw) are highly correlated to temperature features (reanalysis_air_temp_k, reanalysis_avg_temp_k). Total cases from both cities are related to the weather (described by features including average temperature and humidity), however, different from our expectation, aren't highly correlated to any of the features that are being examined.

4.3 Time Series Analysis

The first model we were interested in running on our data was the time series analysis. As inspired by the results of trend analysis during the EDA process, we've decided to further explore to see whether the disease has seasonality. Time series analysis is a specific way of analyzing a sequence of data points collected over an interval of time. The model can be used for forecasting—predicting future data based on historical data.
In order to examine the number of infection cases in relation to time, we want to convert the month column into a datetime object. This will allow it to programmatically pull time values like the year or month for each record. To do this, we used the Pandas to_datetime() method. We then generated a time series plot using Seaborn and Matplotlib, which allowed us to visualize the data.





Similar to our EDA analysis, we also splitted the data and examined them according to the different cities: sj and iq. We could see that from those two graphs, there isn't a clear total case trend in relation to time. Though the graphs differed from what we previously expected, these did make sense since if we take a look back at the graphs of trends of number of cases for different years in both iq and sj (as shown in section 4.2), we would notice that the number of cases does vary a lot depending on the year and the two graphs that we just got further showed that the number of total cases weren't highly correlated to the year we are looking at.

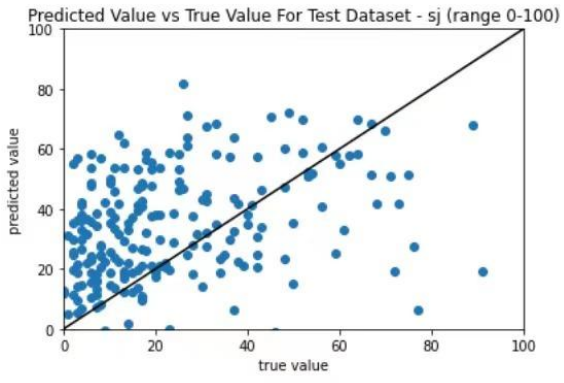
We realized that though there exists some months during the years that have similar patterns like all having the peaking feature, the overall patterns weren't clear enough to use time as the only covariant to predict future cases. After running the time series analysis as the first try, it wasn't surprising that the predictions deviated from true values a lot without even reasonable trends. We have to seek other models which take other environmental and periodic features into consideration for better performance.

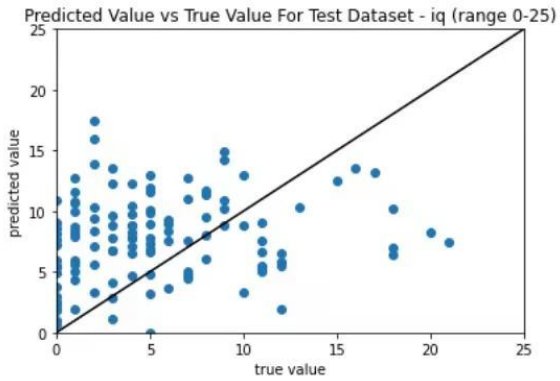
4.4 Regression

The next approach we tried was to implement different regression methods. We decided on implementing regressions since those could help predict the value of the dependent variable in order to estimate the effect of some explanatory variable on the dependent variable. Specifically saying, since we've created heatmaps for the two cities previously and had a gauge of the correlations between each feature, we want to further explore the impact of the different features using regressions.

4.4.1 OLS

We first trained a linear regression using OLS on the total cases and city features with one hot encoding for nominal features. We implemented OLS since it's considered as a common technique for estimating coefficients of linear regression equations which describe the relationship between one or more independent variables and a dependent variable.

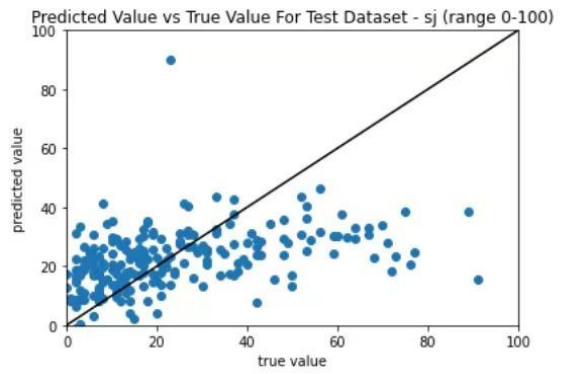


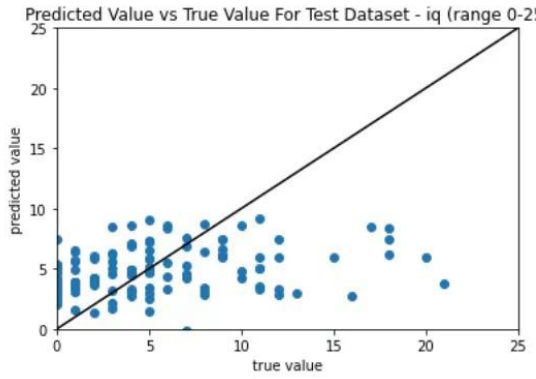


We applied the model onto both cities. And the above two graphs show the predicted values by the model versus the true value of the dataset. We can see that the model didn't perform well for predicting, we then tried to figure out why this model wasn't as successful as we previously expected. We looked at the generated result and noticed its small eigenvalues for both sj and iq. The reason why we are using eigenvalues to observe is due to the fact that an eigenvalue is a number which represents how much variance there is in the data in that direction. The lower the eigenvalue, the lower the variance is within the matrix and higher the chance of high collinearity in the data. As the results of the model shown: The iq data set had the smallest eigenvalue as 1.25e-25. and the sj data set had the smallest eigenvalue as 1.98e-25. Our interpretation and hypothesis of the results: For iq, its smallest eigenvalue might indicate that there are strong multicollinearity problems or that the design matrix is singular. Both standard Errors in the two outcomes assume that the covariance matrix of the errors is correctly specified. For sj, its smallest eigenvalue is 5.78e-26) which might indicate that there are strong multicollinearity problems or that the design matrix is singular.
This result made sense since multicollinearity is the occurrence of high intercorrelations among two or more independent variables, this corresponded to the previous heatmap that we've made where we've observed that total cases from both cities are, different from our expectation, aren't solely correlated to any of the features that are being examined.

4.4.2 Huber Regression

We noticed that from both graphs in EDA and results from OLS, that there are some noise and outliers introduced in the dataset. Thus we wanted to implement a regression model that is robust to outliers. We chose huber regression to achieve this effect since it is less sensitive to outliers in data than the squared error loss. Huber regression is used widely for robust regression problems where outliers are present that can degrade the performance and accuracy of least-square based regressions.

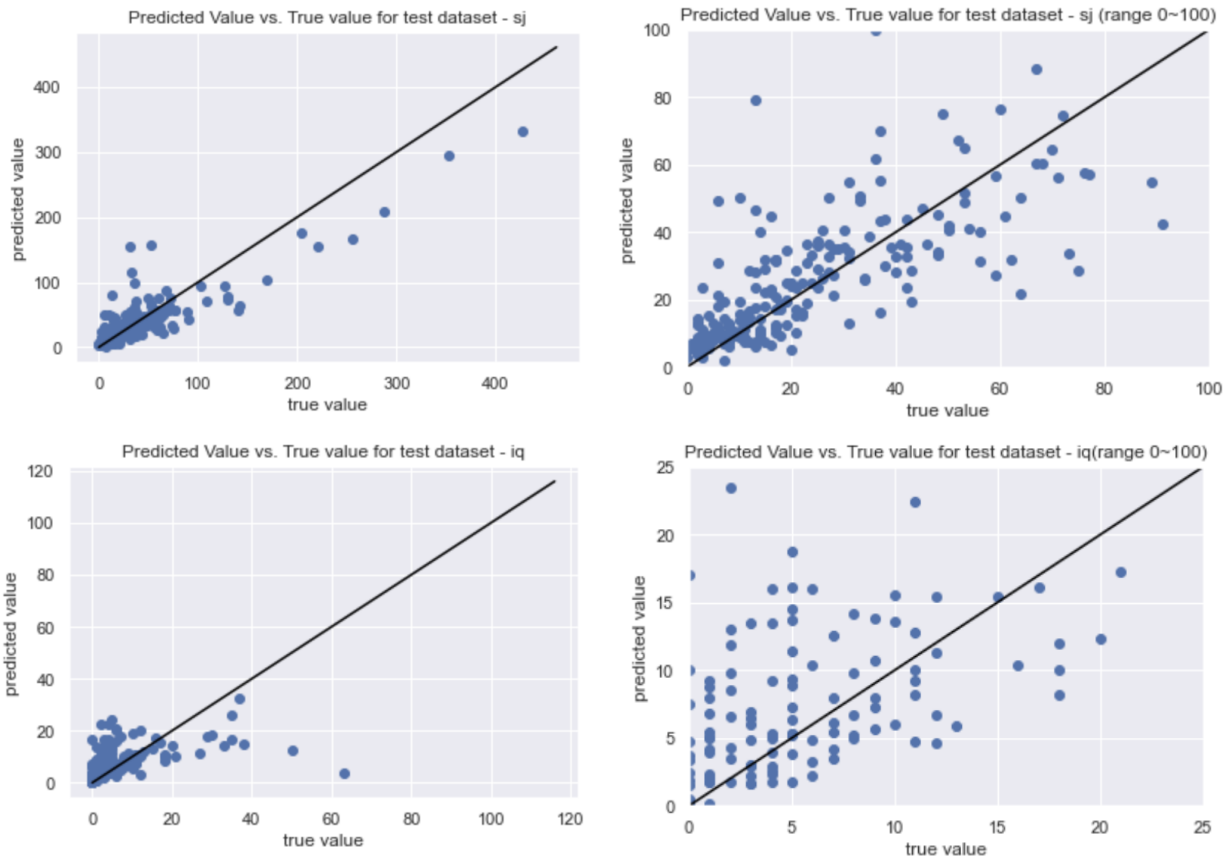




We generated the above two graphs by running the Huber Regression model. We can see that huber regression, compared to OLS, was not impacted much by the outliers and thus provided with a more accurate result.

4.5 Random Forest

Another model we've decided to implement for the problem was a random forest. We decided on implementing a random forest because they can be used for prediction problems, prevent overfitting, gain greater accuracy and give estimates on variable importance. Since a random forest is developed by using a multitude of decision trees and then deciding on the class of input based on the mode of the classes predicted by the multiple decision trees, those many decision trees that have been used helped to prevent overfitting. We implemented our forest so that it would create 70 decision trees at each node to help prevent overfitting the training data. After implementing the random forest model, we used it to predict the outcomes of cases and compared it with the true values. We then plotted the outcomes in the same graphs.



In the four above graphs, the right graphs are zoomed-in versions of the left. We concluded that, with the number of estimators being 15, the predicted value by using random forest for both San Juan and Iquitos were pretty accurate. And the predictions are more accurate for the cases where the true number of cases are smaller. That is possibly because the model is predicting more in a general sense, where the sudden outbreaks are harder for the model to predict. To show the accuracy in detail, we further performed gradient boosting. We examined the most common form of GBM that optimizes the mean squared error (MSE) which is the average of the square of the difference between the true targets and the predicted values from a set of observations, such as a training or validation set. We chose this approach since optimizing a model according

to MSE makes it chase outliers because squaring the difference between targets and predicted values emphasizes extreme values. We got the results that the gradient boosted **MSE for San Juan is 13.6** and **the gradient boosted MSE is 5.6**, further validating our previous conclusion.

5 Conclusion and Future Work

In this paper, we have studied the performance of different learning algorithms on the data of dengue cases in different cities at different time points. The goal of our work was to identify a suitable model that can predict the total number of cases more accurately. In general, we derived two models that include all the predictors in the dataset, and one model(time series analysis) that uses only the time information. Plus, we employed multiple criteria and visually inspected the data to assess the performance of each model. We finally draw the conclusion that the Random Forest method is relatively the most accurate model with the lowest test error rate.

However, the comparison among test errors of each model could only provide us a general idea of the optimal model to choose. Therefore, more benchmarks could be used in the model performance evaluation. We could have done cross validation for each model and compared the results. Though we have got a decent accuracy for this dataset, the accuracy can be improved by using other deep learning methods. Besides, more data sets collected in other time periods can also be included, which could lead to an increase in the accuracy as we have experimented with a very diversified and sparse dataset.

In order to improve the random forest model, we can try bagging and boosting to see if those techniques provide us with a better model. By using bagging, we can get a less complex decision boundary, which could prevent overfitting from happening. By using boosting, we increase the model complexity and aim to decrease the bias of the model. As for the regression models the predictors are very likely to be correlated with each other, which would affect the performance of the predicting models. Therefore, methods like predictor normalization can be applied to the data in order to increase the performance.