DengAI

CIS 520 Final Project

Dinesh Jagai
Department of Computer Science
University of Pennsylvania
Philadelphia, PA
dinesh97@seas.upenn.edu

Pranav Panganamamula
Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA
ppranav@seas.upenn.edu

Julian P. Schnitzler
Department of Computer Science
University of Pennsylvania
Philadelphia, PA
schnitzl@seas.upenn.edu

December 10, 2019

Abstract

The dengue fever is a threatening disease that causes thousands of deaths every year. Using Machine Learning approaches, we predict the future numbers of cases of Dengue fever in two cities of Peru, namely San Juan and Iquitos. Related Work shows that there are many possible models to predict these number, mostly with simpler regression models. Using models like ElasticNet, RandomForests and GradientBoosting, we obtain cross validation errors of at most around 30 MAE. However, it turns out that the test error for RandomForests and GradientBoosting on the hidden test set is significantly higher, whereas the values for ElasticNet stay more or less the same, probably due to overfitting of the latter 2. We finally obtained our best MAE on the test set with a value of 33.69. Finally, we conclude that Iquitos in general has less cases than San Juan, and therefore also has a lower MAE compared between the same models. In future work, one could try to use time sensitive models for this approach, as the goal is to predict the future given the past.

***K*eywords** Machine Learning · Feature Selection · Regression · Data Imputation

1 Motivation

While growing up in the Caribbean, the most feared disease was that of dengue. Dengue is a viral infection transmitted through the bite of an infected Aedes mosquito. Although most patients with dengue will recover spontaneously, a small number will develop more severe life-threatening forms of the disease. Dengue fever is a mosquito-borne disease that occurs in tropical and sub-tropical parts of the world. In mild cases, symptoms are similar to the flu: fever, rash, and muscle and joint pain. In severe cases, dengue fever can cause severe bleeding, low blood pressure, and even death. Because it is carried by mosquitoes, the transmission dynamics of dengue are related to climate variables such as temperature and precipitation. Although the relationship to climate is complex, a growing number of scientists argue that climate change is likely to produce distributional shifts that will have significant public health implications worldwide.In recent years dengue fever has been spreading. Historically, the disease has been most prevalent in Southeast Asia and the Pacific islands. These days many of the nearly half billion cases per year are occurring in Latin America.

Our goal is to predict the number of dengue cases each week (in each location) based on environmental variables describing changes in temperature, precipitation, vegetation, and more. Predicting the number of cases in these cities can be very helpful to support planning of the distribution of pharmaceuticals and other actions to cure people or prevent people from getting ill. Dengue is a significant yet sometimes neglected disease, affecting an estimated $50\breve{}100$ million people per year and costing nearly \$9 billion globally. In Peru there were more than $27,000$ reported cases in 2016, and over $26,000$ reported cases in 2010 in Puerto Rico. Machine Learning will help us to find the relevant features and generate responsible predictions in these two cities. As such, it would help public health experts properly diagnose the dengue problem, integrate solutions to combat it and consequently manage and control the epidemic.

2 Related Work

The general work of using Machine Learning to predict the number of number of cases for diseases over a period of time is vast. For example, [1] discusses disease prediction in general by Machine Learning Over Big Data From Healthcare Communities using a CNN-based multi-modal disease risk prediction. They experiment the prediction models over real-life hospital data collected from central China in $2013 - 2015$. The prediction accuracy of their proposed algorithm reaches 94.8% with a relatively high convergence speed. One of the most common disease that researchers have tried to predict using machine learning is that of Dengue.

Firstly, this [2], paper discusses how to predict dengue outbreaks based on disease surveillance, meteorological and socioeconomic data - it uses a Quasi-Poisson regression in which the variance of count data (dengue counts) is assumed to be a linear function of the mean for to predict the dengue cases. Interesting, they determined that for the predicting dengue outbreaks within a district, the influence of dengue incidences and socioeconomic data from the surrounding districts is statistically significant. This method is advantageous because it doesn't use much computational power and it is a relatively simple method to implement. However, it dengue counts is not necessarily a linear function of the mean and as such, this method has a high variance.

Furthermore, B. Connor [3], in an article highlighting how to predict dengue using different regression methods, predicts the number of cases of dengue in two cities over a five year period. It is imperative to note that inspiration from our project came from Connor's work. His methods again were advantageous in that he used multiple regression methods and found the model that worked best as opposed to the previous paper where only one model was utilized. Moreover, the authors of [4] looked at developing a Dengue Possibility Forecasting Model using Machine Learning Algorithms. Specifically, they use a Gradient Boosting Regression ensemble method to predict the possibility of a dengue outbreak taking place. The use of Gradient Boosting Regression is advantageous here in that it gives lots of flexibility - can optimize on different loss functions and provides several hyper-parameter tuning options that make the function fit very flexible. Disadvantage is that, the Gradient Boosting Regression overfitted the data and lead to high loss values in certain cases.

Also, [5] examines predicting the number of dengue cases in China using several ML techniques including the support vector regression **(SVR)** algorithm, step-down linear regression model, gradient boosted regression tree algorithm **(GBM)**, negative binomial regression model **(NBM),** least absolute shrinkage and selection operator**(LASSO)**, linear regression model and generalized additive model **(GAM)**. These were used as candidate models to predict dengue incidence. They found that the (support vector regression) **SVR** model achieved a superior performance in comparison with other forecasting techniques assessed in this study.The beautiful thing about this paper is the numerous models that they use to predict the number of cases. They compare the advantages and disadvantages of each one. Our team definitely took some inspiration from this paper when it came to implementing many different models and picking the one with the lowest test loss.

Apart from regression methods, this paper [6] focused on the approach of deep learning to predict the number of Dengue cases in Taiwan. This is advantageous in the sense that deep learn and most accurate model the continuous function of the dengue cases. Note that the big disadvantage is this is the amount of data needed. Our team decided to go against deep learning as the number of training samples was a mere 1046.

In addition, due to the time series nature of the problem (predicting the number of cases over a period of time), it is possible to use **LSTM** Neural Nets. This was done by the authors of [7]. Their model was developed from monthly dengue cases and local meteorological data of $2005 - 2018$ among top 20 Chinese cities with a record of the highest dengue incidence. They concluded that the **LSTM** model is beneficial in predicting dengue incidence as compared with other previously published forecasting models.This method was very advantageous as it made adept use of the time series in dengue.

Finally, [8] uses Least Squares Support Vector Machines **(LS-SVM)** in predicting future dengue outbreaks in Malaysia. The data sets used in the undertaken study includes data on dengue cases and rainfall level collected in five districts in Selangor. They found that prediction results of unseen data show that the **LS-SVM** prediction model outperformed the Neural Network model in terms of prediction accuracy and computational time. This was advantageous since it compared a regression model to one using deep learning and actually concluded that the regression model outperformed the deep learning method.

3 Dataset

The dataset used for our problem comes from the DrivenData website for the DengAI competition, and contains climate data for two cities: San Juan, Puerto Rico and Iquitos, Peru. The features given by the data that we consider are:

City and date indicators

- `city` – City abbreviations: sj for San Juan and iq for Iquitos
- `week_start_date` – Date given in yyyy-mm-dd format

NOAA's GHCN daily climate data weather station measurements

- `station_max_temp_c` – Maximum temperature
- `station_min_temp_c` – Minimum temperature
- `station_avg_temp_c` – Average temperature
- `station_precip_mm` – Total precipitation
- `station_diur_temp_rng_c` – Diurnal temperature range

PERSIANN satellite precipitation measurements (0.25x0.25 degree scale)

- `precipitation_amt_mm` – Total precipitation

NOAA's NCEP Climate Forecast System Reanalysis measurement (0.5x0.5 degree scale)

- `reanalysis_sat_precip_amt_mm` – Total precipitation
- `reanalysis_dew_point_temp_k` – Mean dew point temperature
- `reanalysis_air_temp_k` – Mean air temperature
- `reanalysis_relative_humidity_percent` – Mean relative humidity
- `reanalysis_specific_humidity_g_per_kg` – Mean specific humidity
- `reanalysis_precip_amt_kg_per_m2` – Total precipitation
- `reanalysis_max_air_temp_k` – Maximum air temperature
- `reanalysis_min_air_temp_k` – Minimum air temperature
- `reanalysis_avg_temp_k` – Average air temperature
- `reanalysis_tdtr_k` – Diurnal temperature range

Satellite vegetation - Normalized difference vegetation index (NDVI) - NOAA's CDR Normalized Difference Vegetation Index (0.5x0.5 degree scale) measurements

- `ndvi_se` – Pixel southeast of city centroid
- `ndvi_sw` – Pixel southwest of city centroid
- `ndvi_ne` – Pixel northeast of city centroid
- `ndvi_nw` – Pixel northwest of city centroid

The features starting with 'station' were measured by NOAA's GHCN daily climate data weather station. The features starting with 'reanalysis' were measured by the NOAA's NCEP Climate Forecast System Reanalysis. The features starting with 'ndvi' represent the normalized difference vegetation indices of the regions around the city.

We use all of the listed features and disregard additional features given such as year, week of year, and starting date As such, total, we ended up using $20, (p = 20)$ features. There are $936, n_{sj}$ training samples for the city of San Juan and $521, n_{iq}$, training samples for the city of Iquitos. There are $260, ntest_{sj}$, testing samples for the city of San Juan and $156, ntest_{iq}$, testing samples for Iquitos.

For data pre-processing, we changed all temperature based features to the Kelvin scale, and performed imputation to eliminate missing values. We performed **both** mean imputation and regression-based imputation.

To visualize the data, we computed correlation matrices of the features of the training data of both cities and the labels. First we joined the Training data to labels (number of Dengue cases per week) and then computed the correlation matrix using The features are listed along the axes of the matrices in order of their appearance in the list of features given earlier. This allowed us to determine the features that had the highest correlation to the number of Dengue Case y (Test set) and thus the features to keep, to feature the number of dimensions and the possibility of over-fitting.

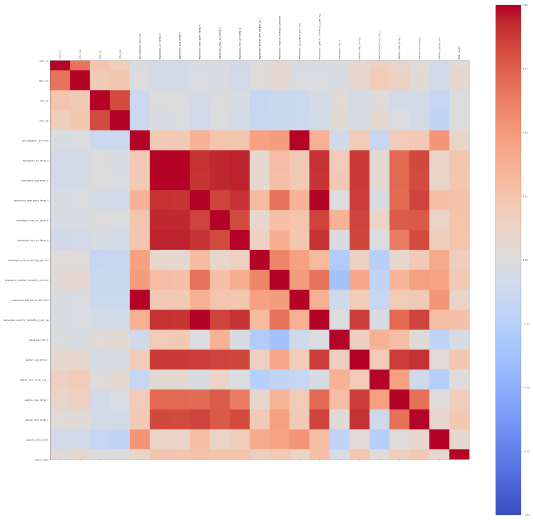

Figure 1: San Juan Feature Correlation Matrix

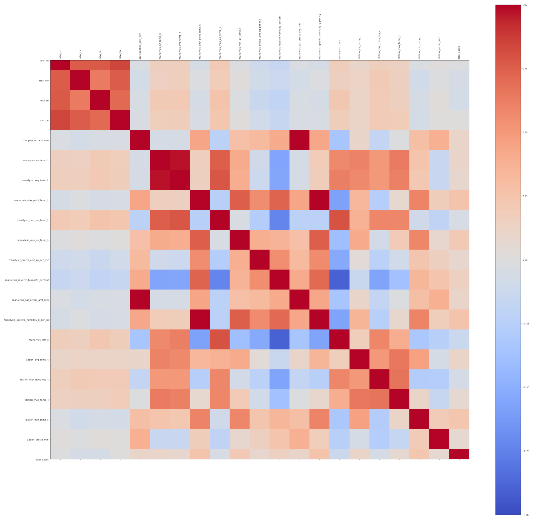

Figure 2: Iquitos Feature Correlation Matrix

We also plotted each feature's correlation with the total number of cases reported using our mean imputed data, for both cities

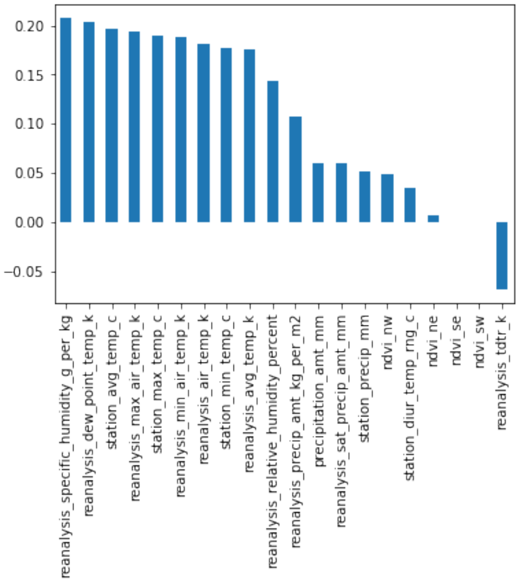

Figure 3: San Juan Feature Correlations with Number of Cases

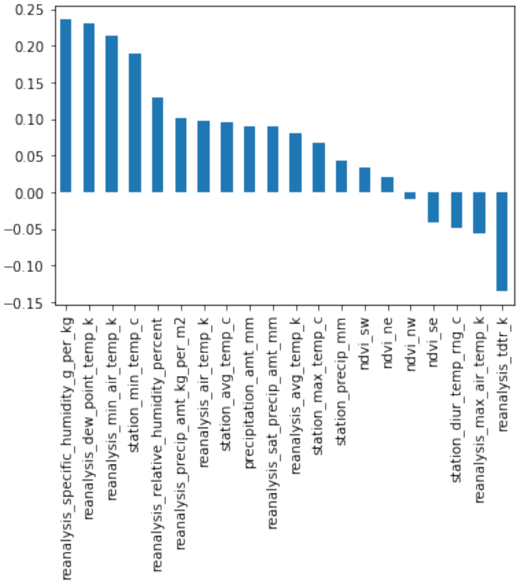

Figure 4: Iquitos Correlations with Number of Cases

Figure 3 and 4 both give us a better visualization of the correlation matrix values to the number of dengue cases as opposed to figures 1 and 2.This again is very helpful in determining the features to keep when we reduce our dimensions. The most important features in both cities are :
`reanalysis_specific_humidity_g_per_kg`
`reanalysis_dew_point_temp_k` `station_avg_temp_c`
`reanalysis_max_air_temp_k`
`station_max_temp_c`
These are shown by the high vertical bars in figures 3 and 4.

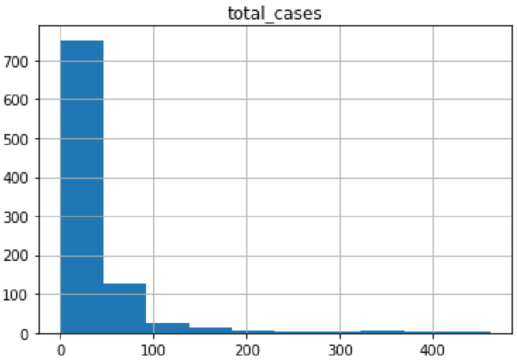

Figure 5: San Juan Total Cases per Week

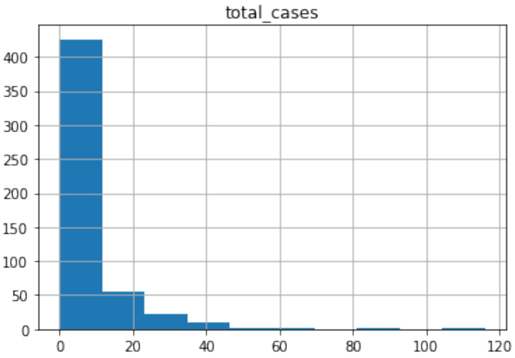

Figure 6: Iquitos Total Cases per Week

These problems show, that the majority of cases are $0 - 100$ for san Juan and $0 - 20$ for Iquitos.
This is very important because is means that our model will tend to output in this range and as such it's distribution will be really similar. This gives us an idea for the type of model to use.

4 Problem Formulation

When formulating the machine learning task for this problem, we concluded that the best representation and implementation of the problem was a supervised learning task that attempts to predict a real-valued output, the total number of dengue fever cases for a given week in a particular city given multiple real-valued inputs, our features such as humidity, temperature, etc. Therefore, our models will utilize regression techniques to predict the number of reported dengue fever cases.

We decided to use Mean Absolute Error (**MAE**) as our main error metric, as it is commonly used in regression based models. It is also much better than the RMSE error when evaluating performance. [9]

To such an extent, our problem is given as -"Predict the number of dengue cases each week in San Juan and Iquitos over a five year period $(2008 - 2013)$ using given environmental variables describing changes in temperature, precipitation, vegetation, and more from 1990"

5 Methods

For our **baseline models**, we wanted to start to approach this problem using simpler regression techniques. Therefore, our baseline methods involved training our models using Ordinary Least Squares Regression (OLS), Ridge Regression, and Lasso Regression. We chose these three as they are relatively simple models, with OLS being the simplest and Ridge and Lasso imposing some regularization penalties to reduce any over-fitting.

In addition to these baseline models, we also utilized several more advanced models to get a more accurate estimate. These include elastic net regression, kernel regression, random forests regression, gradient boosting, and bagging regression. We discuss each of these models below, and which ones showed the best results in the Experiments and Results section of the report.

One of the models we chose was Random Forests regression, which is one of the most effective machine learning models out there. Random forest models use ensemble learning and as such make predictions by combining predictions from a sequence of base decision trees to get a stronger prediction. The base decision tree models are constructed independently by taking random subsamples of the input training data. We decided on using random forest regression as one of our more advanced models due to the fact that random forest models are good at handling tabular data with numerical features and they can capture non-linear interactions between our features and the target output, which is important because it is likely that the relationships between the features and the total number of cases is non-linear.

Another model we chose to experiment with was Gradient Boosted regression trees, another highly effective machine learning model. Gradient boosted regression trees are similar to random forests regression in that they both utilize ensemble learning and use decision trees as the base models. However, while random forest models construct base classifiers independently with subsampling, gradient boosted trees use gradient boosting. Gradient boosted regression trees have similar advantages for our problem as random forests regression.

We also chose to utilize elastic net regression, which is a regression method that uses a regularization technique that combines L_1 and L_2 penalties. Since our dataset is somewhat small, this technique is useful to observe any possible effects of different regularization penalties on our predictions and to reduce any possible overfitting, which is a major concern for this problem.

We also debated on using neural networks and deep learning for this problem, but ultimately decided against it since our data-set wasn't large enough to use a neural network for training and that different regression methods would instead be more effective.

In terms of implementation, we mainly relied on the methods offered by sklearn, including some automatic hyperparameter search with CV. For pre-processing, we generated new CSV files with imputed values to not always need to regenerate the data. We then read in the data into pandas DataFrames, converted them to numpy arrays and trained our sklearn models. We then saved the obtained models with pickle on our google drive. The evaluation is then mainly done with sklearns kFold class. Moreover, we could save the predictions for the test set in another CSV file and upload it, which gives us the possibility to receive the MAE for the hidden test set on the website.

6 Experiments and Results

For evaluation, we decided on the following framework:

During our process of generating our estimation of the testing error, which in our case is 5-fold cross validation, we saved all 5 trained models from these process for later use.

To then evaluate the model, we then just take the 5-fold CV error, i.e. the average MAE obtained through all 5 trained models.

Whenever possible, we used the predefined CV methods from sklearn to automatically optimize our hyperparameters in a given interval, using cross-validation.

Another way to evaluate is through a hidden test set on the website, which takes as input our predicted values for the test set and simply returns the MAE.

As our performance matrix, we decided on the MAE and ran 5-fold Cross Validation to estimate the error on a testing set and search for the optimal hyperparameters.

MAE is calculated by:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$$

The MAE in general is a good choice for time series data, like the one we have here.
Another point in choosing MAE as the performance metric for us was that it can is very easily interpreted. A MAE of x

simply means that a value is on average of the predicted value by about x. This helps a lot in imagining what might be good or bad values on a test set.
As our dataset contains of data from two different cities, we decided to train two different models, one for each city, and also compare the MAE that we receive from evaluating both.
Moreover, we decided to always compare both the mean imputed and regression imputed data, to be able to observe the influence of both methods on our performance. In addition we also attempted building an ElasticNet model with only the five most significant features found in our visualizations of the data:
`reanalysis_specific_humidity_g_per_kg`
`reanalysis_dew_point_temp_k`
`station_avg_temp_c`
`reanalysis_max_air_temp_k`
`station_max_temp_c`

Table 1: Experimental results (MAE) on different methods

		Mean imputed		Regression imputed	
Function	hyperparameters	San Juan	Iquitos	San Juan	Iquitos
Linear regression, OLS	-	29.17	6.84	28.89	6.82
RidgeRegressionCV,	$\alpha \in [0, 1]$	28.92	6.81	28.84	6.79
LassoRegressionCV,	$\alpha \in [0, 1]$	28.77	6.81	28.76	6.83
ElasticNet regression,	$\alpha \in [0, 1], l1\_ratio \in [.1, .5, .7, .9, .95, .99, 1]$	28.78	6.82	28.79	6.82
ElasticNet with 5 features,	$\alpha \in [0, 1], l1\_ratio \in [.1, .5, .7, .9, .95, .99, 1]$	28.75	6.80	28.42	6.62
RandomForest	$n\_estimators = 100$	32.90	7.60	33.30	7.68
RandomForest	$n\_estimators = 1000$	32.66	7.68	33.01	7.68
RandomForest	$n\_estimators = 33, max\_depth = 20$	33.16	7.78	33.15	7.72
GradientBoosting	$\alpha = 0.9, learning\_rate = 0.1, max\_depth = 3$	31.10	7.64	31.88	7.61
BaggingTrees	$n\_estimators = 16$	33.04	7.54	33.45	7.58

Testing on the test set on the website gave us:

Table 2: Experimental results (MAE) on hidden test set

Function	hyperparameters	MAE
ElasticNet regression,	$\alpha \in [0, 1], l1\_ratio \in [.1, .5, .7, .9, .95, .99, 1]$	33.69
RandomForest	$n\_estimators = 33, max\_depth = 20$	79.20
GradientBoosting	$\alpha = 0.9, learning\_rate = 0.1, max\_depth = 3$	61.00

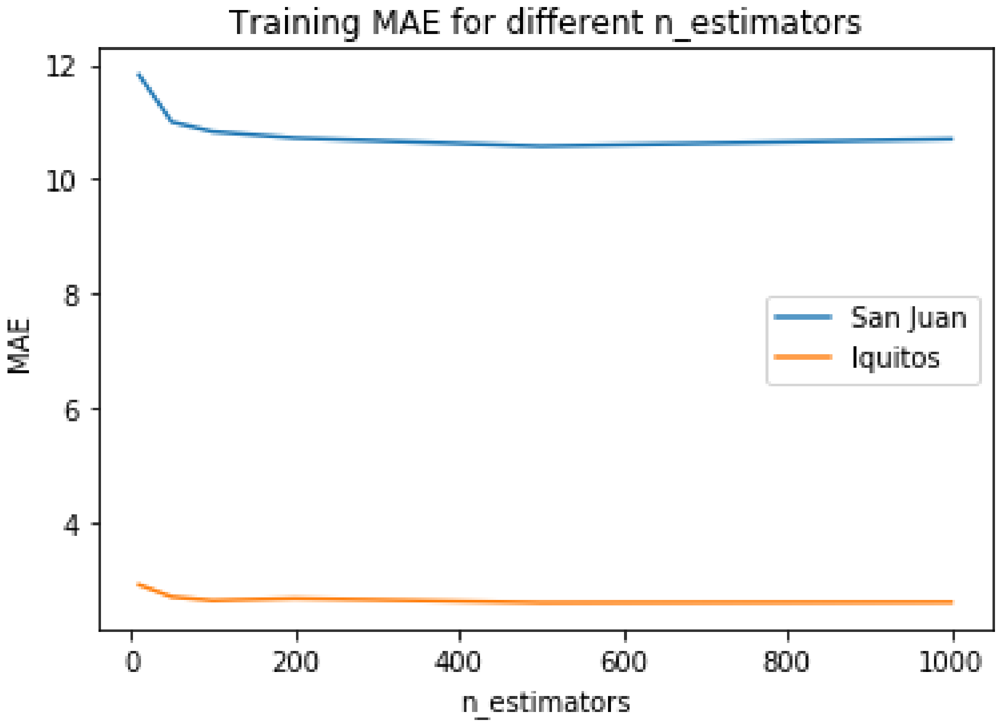

Figure 7: Training MAE for different $n\_estimators$ in RandomForest

For estimating the $n\_estimators$ in Fig. 7, it turned out we reached the best model for about 33 estimators. For a higher number of estimators, the MAE is not significantly improving, for lower values, the MAE is a lot higher. Similar figures were generated in order to determine the optimal hyper-parameters for the different regression models implemented.

7 Conclusion and Discussion

Across all the methods we evaluated by cross-validation, both ElasticNet and Lasso regression performed the best in terms of cross-validation. As the MAE values of ElasticNet are very close to Lasso, we assume that the $l1\_ratio$ hyperparameter is very close to 1. In terms of cross-validation error, we also observe that the simpler regression models perfrom better than more complicated models like RandomForests and GradientTreeBoosting.
We assume that one of our main problems is overfitting. All of our cross validation errors are at most around 30, while the error on the hidden test set reaches this level for ElasticNet, so the regression models seem not to have that much of a problem with overfitting, but RandomForests and GradientBoosting perform very weak, getting 79 respectively 61 average error, meaning they are on average almost 70 wrong. In addition, as sklearn is usually optimizing MSE or another squared loss instead of MAE, this might also explain our weak performance.

Another interesting point to observe is the difference between MAE for San Juan and Iquitos. The general MAE is always about 4 times smaller for Iquitos than for San Juan. Observing the visualization of the number of cases in Fig. 5 and 6, we found out that San Juan reaches values of around 400 cases per week, while Inquitos only reaches values of at most 200 cases per week. This difference might explain why the MAEs for both cities differ by almost factor 4.

The problem in general seems very likely to generate models that highly overfit on the data. As there are only 1500 training samples, of which only 750 have no missing values, we highly tend to overfit, even with imputed data. Moreover, there are some very similar rows. Trying to fit KernelRegression models to our data led to many problems with singular matrices, even with PCA as pre-processing beforehand. We observed some rows that were very similar to each other and probably fostered overfitting.

Over the course of this project, we learned many things. One of the lessons we learned was the importance of visualizing our data. Putting in the effort to generate visualizations of our data really helped us understand the nature of the data and how it might contextually relate to the problem at hand. For example, plotting the graph of each feature's correlation to the output number of reported cases of dengue fever gave us a greater understanding of which of our features were more closely related to the output label. In addition we also gained a greater understanding of how machine learning techniques can be used in practice in relation to real world problems.

Future work on this problem could involve factoring the progression of time into our models, as currently we drop features with information about time such as year, date, etc. from the dataset. Another possible extension of this project could be creating an online implementation that would continuously take in data and could be used to continually predict new cases, which would be useful in preventing and mitigating future outbreaks. If we had more training samples, an optimal solution would be to implement an **LSTM** as done in [7].This method will be very advantageous as it will make adept use of the time series in dengue.

References

[1] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang. Disease prediction by machine learning over big data from healthcare communities. *IEEE Access*, 5:8869–8879, 2017.

[2] Raghvendra Jain, Sra Sontisirikit, Sopon Iamsirithaworn, and Helmut Prendinger. Prediction of dengue outbreaks based on disease surveillance, meteorological and socio-economic data. *BMC infectious diseases*, 19(1):272, 2019.

[3] Brian Connor. Dengue fever and how to predict it, Nov 2018.

[4] P Muhilthini, BS Meenakshi, SL Lekha, and ST Santhanalakshmi. Dengue possibility forecasting model using machine learning algorithms. 2018.

[5] Pi Guo, Tao Liu, Qin Zhang, Li Wang, Jianpeng Xiao, Qingying Zhang, Ganfeng Luo, Zhihao Li, Jianfeng He, Yonghui Zhang, and Wenjun Ma. Developing a dengue forecast model using machine learning: A case study in china. *PLOS Neglected Tropical Diseases*, 11(10):1–22, 10 2017.

[6] Sumiko Anno, Takeshi Hara, Hiroki Kai, Yi Chang, Ming-An Lee, Kei Oyoshi, Yosei Mizukami, and Takeo Tadono. Environmental science & technology 2018. *Expert Review of Molecular Diagnostics*, 10(8):987–991, 2018.

[7] Jiucheng Xu, Keqiang Xu, Zhichao Li, Taotian Tu, Lei Xu, and Qiyong Liu. Developing a dengue forecast model using long short term memory neural networks method. *bioRxiv*, 2019.

[8] Yuhanis Yusof and Zuriani Mustaffa. Dengue outbreak prediction: A least squares support vector machines approach. *International Journal of Computer Theory and Engineering*, 3:489–493, 01 2011.

[9] Kenji Matsuura Cort J. Willmott. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Inter-Research Science Publisher*, 30, No. 1., 01 2011.