Dauphine | PSL
UNIVERSITÉ PARIS

Data Project
Master Digital Economics – Madalina OLTEANU

Predicting Influenza epidemic: A Machine Learning approach

Kathleen ROGAN

&

Jinane Amal

Abstract

Infectious diseases carry significant human and economic costs, making accurate surveillance methods and swift political responses crucial. Influenza, a well-known seasonal virus, is a prime example. A major challenge with the flu is its ability to evolve, becoming more resistant to vaccines each year, which poses a greater risk to vulnerable populations. The aim of our project is to develop effective models for predicting influenza outbreaks using various machine learning techniques, including Logistic Regression, Random Forest, Gradient Boosting, ARX and a multivariate LSTM; and heterogeneous data sources.

Introduction

0.0.1 Related Works

The idea of predicting flu epidemic using search queries was first introduced in 2006 by Eysenbach, who demonstrated a positive correlation between searches on Google and epidemic data. [Eysenbach(2006)]

Subsequent research using machine learning has relied on data from Twitter[Achrekar et al.(2011)] (now X). These studies showed that that the number of tweets related to influenza were positively correlated with influenza-like illness (ILI) cases reported by the Centers for Disease Control and Prevention (CDC). The research also relied on Google Flu Trends[RE;(2013)], which was found to be statistically significant to predicting flu outbreaks.

In contrast, most research involving Wikipedia in the health sector have focused on assessing the accuracy of its information for public use. Only few studies have explored influenza prediction using Wikipedia data, and those that have typically rely on statistical methods rather than machine learning.

Given this, we believe it would be valuable to explore predicting seasonal influenza outbreaks using online information that would be freely available.

0.0.2 Goal of the project

Our project consist of a predictive analysis, using freely available data on the internet to predict influenza cases. The main objective of this project is to see if heterogeneous data, such as weather, Google trend, and Wikipedia page views, can be used to predict flu infections and thus predict Influenza epidemics in advance. This would prove beneficial for countries with limited resources, enabling them to put in place preventive measures and reduce the impact of flu on the more vulnerable populations.

For our analysis, we tested three commonly used machine learning models, Linear Regression, Random Forest, and Gradient Boosting. Additionally, we also evaluated a time series ARX model, which is commonly used in epidemiology in the predictive analysis of Influenza, and employed a multivariate Long Short-Term Memory (LSTM) deep learning model.

We trained the different models on the learning set and, based on the results we identified the best model. We then ran the best model on a validation set to see how it performed on unseen data. Finally, we compared the predicted results we obtain with a test set and the actual outcomes.

Our project focuses on the time period from 2008 to 2020. This range was chosen for two reasons. Firstly, Wikipedia's data dump fully covered started in 2008, so to align all our datasets we used the starting date from our Wikipedia dataset. Secondly, we ended our dataset on December 2019 due to COVID-19 pandemic. The first confirmed case of COVID-19 in the USA was recorded on the 18th of January 2020. After 2020 the data is not heavily impacted, as only a few cases were misclassified into ILI. However since 2022 we have seen an notable increase in ILI cases and this is mainly due to the facts that after the lockdown immune system were weaker so more people had the flu but also

less people tested for COVID which means that a majority of people that actually had COVID and went to the doctor were reported having flu symptoms.

0.0.3 Difficulties encountered

We decided to exclude tweets as a feature due to changes in X's API policies, which have limited the ability to freely extract tweets based on keywords and made it to complicated for us to use. Additionally, Google Flu Trends was discontinued in 2015 due to its lack of accuracy. At best, it could predict outbreaks only a few days in advance, but often struggled with overfitting, leading to predictions that were much higher than the actual results.

Instead, we used Google Trend. However, it is important to note that between periods frequencies of visits are not normalized the same way, as the numbers of search queries fluctuates which can impact the end results if it is not taken into account.

Another issue we encountered was with the availability of Wikipedia Pageview data. Currently, the data is only available starting in 2015. Before that there is a Wikipedia dump of discontinued services which includes page views for all the pages. It consists of all pageviews for all articles on Wikipedia grouped by hours between 2007 to 2016 which makes the extraction of weekly information particularly challenging.

Initially, we intended to include vaccine coverage in the state of California as the vaccination rate can have an impact on the way the virus spread.
However, upon reviewing different datasets available both from the CDC and the Californian Health department, we found significant data discrepancies. For example, for the year 2024 some of the information was in reality from the year 2018. Moreover, the available data was monthly data which would have greatly reduced the granularity of our study. Consequently, we decided to not use the vaccine datasets for our analysis. All the steps applied on the vaccine data are still available in the Jupyter notebook.

Throughout our project, we also had to account for the fact that our data was time series. This means that the ordering of the data matters. It also implies that when you have multiple data sources you have to be extremely careful when merging them so that the time steps are aligned and are the same.

I. Datasets and Preprocessing

1.1 Demographic

Previous studies have demonstrated that population density has a direct impact on the severity of epidemics, with higher initial densities leading to larger outbreaks. For this reason, our analysis includes demographic density data for the United States.

We are using the Annual Estimates of the Population for the U.S. per States, as well as Puerto Rico, provided by the Federal Reserve Bank of St. Louis, which has census records for all 50 states from 1900 to 2023. [US(2024)]

These individual census records will are combined to calculate overall population density in the U.S. and identify the state that may be more prone to faster disease spread due to high population density. In our case we limited our study to the state of California.

It has the largest population and one of the largest density of population per miles is California. See Figure 5 and Figure 6 in the annex for visual representation of this.

1.2 Influenza

For our target variable, we use the number of cases reported for Inflenza-like illness (ILI). Since accessing real-time data from state hospitals can be complex, we decided to use data from the Centers for Disease Control and Prevention (CDC). The CDC provides comprehensive weekly U.S. surveillance reports, which in 2023 included data from 3,400 public and private healthcare providers across 50 states and over 110 million patient visits. However this data was only available from 2001 to 2020.[Fluview(2024)]
To complete it, we also used the ILI reports from the California Health and Human Services (CHHS), which ranged from 2010 to 2024. [Departement of California(2024)]

Some discrepancies were observed mainly due to the fact that the CHHS does not necessarily obtain data from private clinics on a regular bases compared to the CDC. These are observable in Figure 7 of the annex.

1.3 Wikipedia

Wikipedia provides information on more than 185 topics and is used by 23 billion viewers in the world per month. It has a freely accessible tool called Pageview Analysis [Wikitools(2024)]that allows to measure the visits of a page according to keywords based on users and platform since 2015. In our project, we are using keywords that are linked to influenza, such as its symptoms or the different variants of flu that exist.

To select the pages relevant to our study we used a Page crawler using Breadth-First Search (BFS). The primary Wikipedia article on Influenza served as the root node. All the other articles linked from the Influenza page are considered as the nodes of depth 1,while links from those articles were treated as nodes of depth 2. We limited our search at depth 2, since articles at greater depths were less relevant to our analysis. Additionally, we only considered articles in english. Using BFS, all child nodes of a given node are explored level by level until the tree's depth limit is reached. It allowed us to explore a large number of articles and isolate the most relevant ones to use as features.

We obtained a list of 50 articles and with using Pageview we extracted the number of visits each weeks between 2015 to 2024. For the pageview data before 2015 the scalable solution was to use a custom package in R called Wikipediatrend [Petermeissner(2018)]. It enabled us to extract pageview from the Wikipedia dump for 32 of our main articles between 2008 and 2016 [Wiki page(2007)]. The articles that were not present were dropped from the list. Merging and using different cleaning steps we obtain a final dataset of Wikipedia pageviews per week from 2008 and 2024. The R code for this is available on GitHub. [Rogan(2024)]

1.4 Weather

The reproduction rate of the Influenza virus is highly correlated to weather conditions, as dry and cold weather increase the risk of infection. This is why we decided to take into account different features of weather data. Using an API query, we downloaded historical

daily weather data for the state of California. We then preprocessed the data to obtain a weekly time step in our dataset. [Corporation(2024)]

1.5 Google Trends

Google Trends provides real-time insights into search behavior, making it a valuable tool for predicting trends. For example, the term "flu," which shows clear seasonal spikes, can be used to track patterns over time and compare data across U.S. states. The platform offers free, accessible data that can be easily downloaded as CSV files. For our project, we downloaded search trends for symptoms and keywords related to influenza.

One limitation of Google Trends is its restriction of only five keywords per search. Another main limitation is that Google trends settings do not allow data spanning periods longer than five years to be weekly aggregated. As a result, we manually gathered these search volumes for five-year period for groups of 5 keywords across the period spanning from 2008 to 2024.

To mitigate the effects of varying normalization across different queries, we applied a rolling average over a four-week window. This reduces noise and smooths the data allowing for reliable comparisons between keywords and consequently a robust keyword trends analysis . The obtained datasets were further cleaned and standardized. The datasets were then merged on the Dates column.

All the datasets used and created for our project are available on the GitHub repository. It is important to note that all the cleaning steps are performed in the corresponding Jupyter notebook. [Rogan(2024)]

1.6 Preprocessing

For each datasets we checked the presence of missing values and cleaned the datasets to retain only the relevant information when necessary. For multiple files containing similar information, we merged them into one dataset for ease of manipulation. Some of the datasets were on a daily basis and we had to convert them to a weekly bases by aggregating the data with Saturday as the end of the week. All the features were combined, with the target being the number of ILI cases from the Influenza dataset. To improve the models efficiency the data was scaled. Due to our final dataset being small, with only 626 rows we had to split the data in an unconventional percentage. We split the data into 60% training, and 40% testing to avoid important overfitting.

II. Methodologies

2.1 Linear Regression

The scientist Francis Galton was one of the first to conceptualize linear regression. It is the most used statistical method for relationship modeling. It was then later enhanced by Karl Pearson and Ronald Fisher for both prediction and explanatory purposes.

Linear regression models the relationship as a straight line minimizing the difference between the observed and predicted values by the model.The general formula is as follows:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon \tag{1}$$

We get the β_i coefficients using Ordinary Least Squares (OLS) method which minimizes the sum of squared residuals:

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2}$$

The solution is derived as:

$$\boldsymbol{\beta} = (X^T X)^{-1} X^T y \tag{3}$$

where X is the design matrix of independent variables, and y is the vector of observed values.

Linear regression is simple to implement and interpret, especially for smaller datasets . However, the linearity assumption may not always be true, when high correlated independent variables affect the coefficients estimations this creates multicellularity not to mention that the model is sensitive to outliers and prone to over-fitting. [Aldrich(2005)]

2.2 Random Forest

Popularized in 2001 by Leo Breiman and Adele Cutler, Random forest is an ensemble method which combines multiple decision trees each trained on a random subset of the data and features that are selected by the model, before aggregating to obtain a single result. The samples are drawn using a bootstrapping method.

For a given input x, the prediction of the Random Forest is computed as follows with T the number of trees and $h_t(x)$ the prediction made for t:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x), \tag{1}$$

The predictions from all trees are aggregated to obtain an average.

Random Forest is a method that reduced overfitting, is flexible and can be used to determine feature importance. However, unlike a simple Decision Tree it takes more time, requires more resources and is more complex. [Breiman(2001)]

2.3 Gradient Boosting

Introduced in 2001 by Jerome Friedman, Gradient Boosting is a machine learning technique that builds a prediction model by combining the predictions of several weaker models (such as decision trees). It sequentially fits new models to the residual errors made by previous models using gradient descent to minimize the loss function.

The model at iteration stage m is as follows:

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x), \tag{1}$$

with $F_{m-1}(x)$ is the model from the previous iteration, $h_m(x)$ is the weak learner fitted to the residuals and ν is the learning rate, controlling the contribution of $h_m(x)$.

The algorithm requires the model to have a set constant value, which is the mean of the target variable for regression tasks for instance, and the residuals to be computed.

Gradient boosting is robust, provides accuracy, flexibility as it handle various loss functions and clear insight on the feature's contributions. However, it presents certain limitations as it is computationally costly, time consuming for larger datasets, is sensitive to overfitting and requires careful parameter tuning. [Friedman(2001)]

2.4 ARX

The autoregressive exogenous input (ARX) is a time series model that combines an AR model and an X model. The AR model consists of a linear combination of past values and stochastic process of the form:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t \tag{1}$$

Where the the value of the time series at time t depends on its own lags, a constant c, and its error term.

The exogenous input model is the following:

$$Y_t = c + b_1 u(t-1) + \cdots + b_q u(t-q) + \epsilon_t \tag{2}$$

In this case the value of the time series at time t is dependent on a constant, and an exogenous variable b_i.

When combining those two models we obtain the ARX model:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t + b_1 u(t-1) + \cdots + b_q u(t-q) + \epsilon_t \tag{3}$$

This model is used when future values of Y_t are dependent on past values of the same variable and related variables as well. It takes into account time-delay which is especially useful when modeling the evolution of a virus. [Hedengren(2023)]

2.5 Multivariate LSTM

The Long short-term memory (LSTM) model is a deep learning model that uses neural network. It is based on the recurrent neural network (RNN) with the goal of solving the vanishing gradient problem. The information are stored using three gates that decided on the updating and the memory. The first gate is the input gate which is used to decide on the importance of an input relative to the entire data. The input variables are identified using the following equation with w_f the learned weight matrix, h_t the output vector, x_t the input, and b_f the bias vector:

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

The second gate is the forget gate that decides when an information that was put in the memory is considered as new or has already been seen. The following equations are used to enter the forgetting gate with i_t the input gate activation, C'_t the cell input activation vector and C_t the cell state where the memory is updated.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t \times C_{t-1} + i_t \times C'_t \tag{4}$$

And finally, the output gate determines what information in the memory will be in the output.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

The output gate is activated. These equations illustrate the internal structure of the LSTM network. This algorithm is generally use to predict time series events that need to captures dependencies in the long-run as the cell remember over a given period of time. [MH;(2024)]

III. Performances of the Models

3.1 Feature Selection

To see if the selection of certain features had an impact on the results of our models we used two types of process. Firstly, we used the Pearson correlation coefficient (PCC) method which is one of the most used method for linear correlation. It was developed by Karl Pearson based on the idea introduced by F.Galton and correspond to the covariance of variables divided by their standard deviations. It corresponds to a normalized measure of the covariance between -1 and 1. However this technique has some limitations. When the data is noisy it is difficult to extract correlation coefficient.[Pearson(1895)]

Secondly, we used our Random Forest model to determine the feature importance using Gini importance as a default. The R^2 of the model was calculated each time a feature was excluded to observe how it impacted the results. [Breiman(2001)]

The model was used on the most important features according to the PCC, then using the features selected by the Random Forest and finally using all the features.

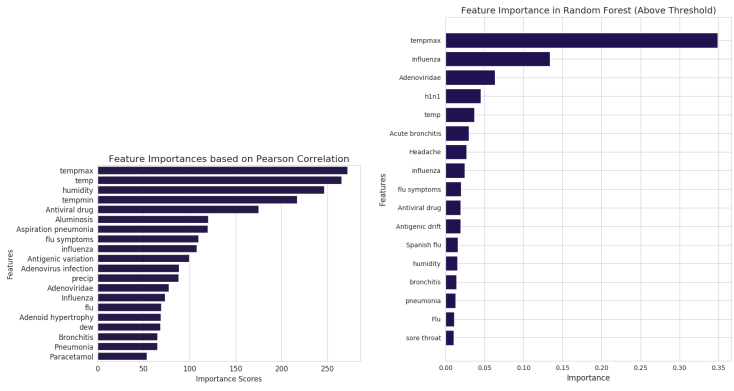

Figure 1: Selection of features using two methods

We can observe in Figure 1 that the Pearson Correlation isolated 20 features and the Random Forest only 17. In both methods the most important feature is the maximum temperature which makes sense as it increases the likelihood of contamination. The rest of the features are similar with only differences in the ranking.

3.2 Selection of the best model

For Linear Regression, Random Forest and Gradient Boosting we used the sklearn models and used GridsearchCV to select the best parameters. Moreover, for our ARX model and our multivariate LSTM we created them using Keras and selected the epoch best on the minimization of the loss function.

Table 1: R^2 VALUES AND PARAMETERS FOR LINEAR REGRESSION, RANDOM FOREST, AND GRADIENT BOOSTING

Features	Linear Regression	Random Forest	Gradient Boosting
All Features			
Parameters	positive=True	max_depth=15	max_depth=2 n_estimators =50
R^2	0.704	0.630	0.768
Pearson Features			
Parameters	positive=True	max_depth=2 min_samples_split = 35	n_estimators=200 learning_rate = 0.01
R^2	0.554	0.600	0.563
Random Forest Features			
Parameters	positive = True fit_intercept=False	max_depth=20	max_depth = 2 n_estimators =50
R^2	0.537	0.626	0.758

From Table 1 we can see that the worst performance is achieved by the Linear regression model with the Random Forest selected features and that in general all the models perform badly with less features. Surprisingly, the Random Forest model had worse performances than expected. This can be explained by the limitation of data during the training period. And the model that performed the best in terms of R^2 out of the three was the Gradient Boosting when using all the features.

Table 2: R^2 VALUES AND PARAMETERS FOR ARX, AND MULTIVARIATE LSTM

Features	ARX	LSTM
All Features		
Parameters	batch = 20 epoch = 280	units = 50 batch = 70

Continued on next page

Features	ARX	LSTM
R^2	0.944	0.789
Pearson Features		
Parameters	batch)=20 epoch = 280	units = 50 batch = 72
R^2	0.756	0.631
Random Forest Features		
Parameters	batch = 20 epoch = 280	units = 50 batch = 72
R^2	0.774	0.773

From Table 2 we can observe that in general, both the ARX and LSTM model perform very well with limited learning data. The ARX has really good results with all the features.

To select our final model, we used the highest R^2 obtained during the learning phase . The best model that was identified was the ARX model incorporating all the features as it had the highest score.

We expect that our model is overfitting and the prediction results will probably less accurate than what we obtained with the learning set.

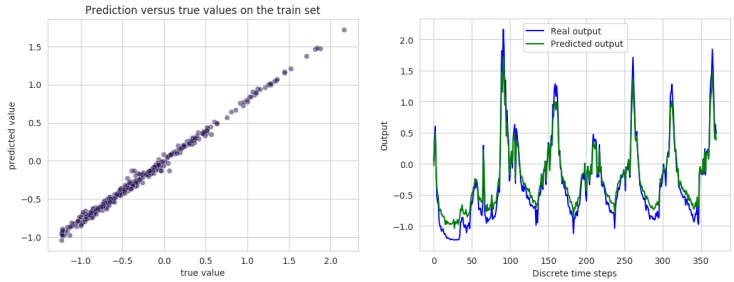

Figure 2: ARX performances on the learning set

As we can see in Figure 2 the ARX model is able to obtain predicted values that follow quite closely the real values of ILI cases.

3.3 Results

When predicting the influenza cases using the test set with our selected model, we confirmed that our model was overfitting and that the R^2 obtained was very low as we can see in Figure **??**. This can be due to multiple factors. The first explanatory factor, is that since 2016 we have seen an increase of Influenza cases, due to the virus being more resistant to the vaccine and less people getting vaccinated. We can clearly see that our model underestimates its prediction of cases. Even when taking into account the increase in population the model is not more accurate in its predictions. And the second factor, is that since the dataset is weekly data and only between 2008 to 2019 there are few observations which can have an impact on the quality of the predictions the model makes. We obtained an R^2 of 0.264 on the whole test set.

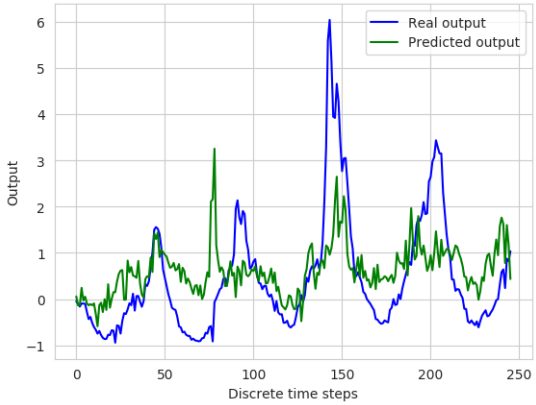

Figure 3: Prediction of ARX model on the test set

However, when we take smaller periods of time, for example over a period of five weeks we obtained an R^2 of 0.72.

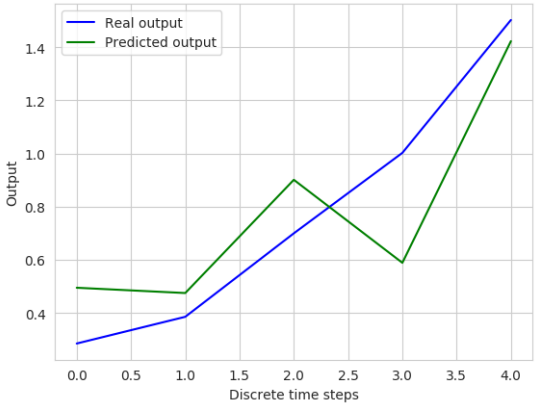

Figure 4: Prediction of ARX model for five weeks

IV. Conclusion

In conclusion predicting the number of Influenza cases is hard, and COVID-19 has made that task even more difficult. However, the model seems to be able to predict the patterns and seasonality of contamination quite accurately. One way we could redirect our analysis, to obtain more precise prediction would be to predict time frames that are more susceptible to see an increase in cases and time periods where there is a decrease.

Appendix

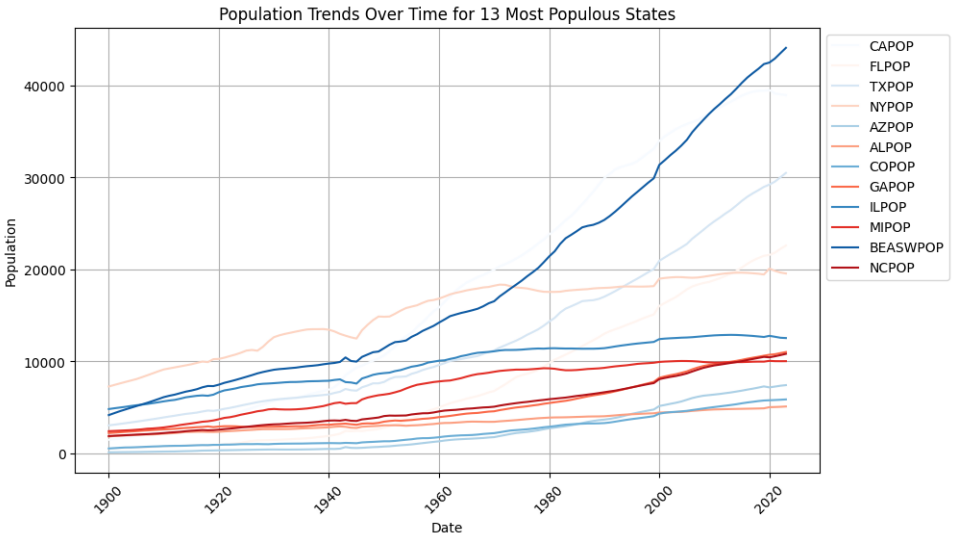

Figure 5: Population Trends in the 13 Most Populous U.S. States

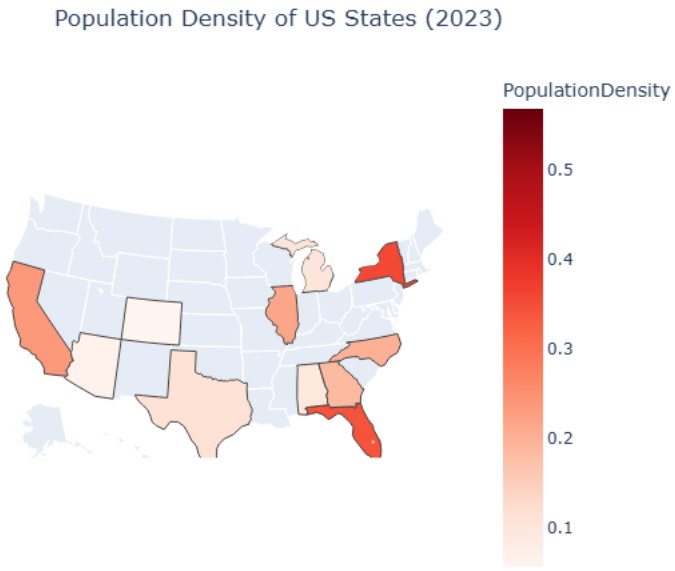

Figure 6: Density of population in the USA

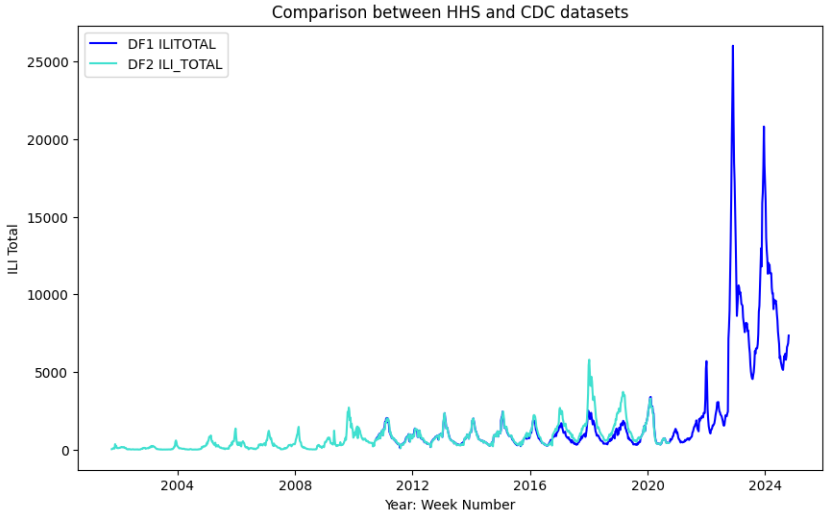

Figure 7: Influenza cases from the CDC and CHHS

References

[Achrekar et al.(2011)] Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. 2011. Predicting Flu Trends using Twitter data. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 702–707. https://doi.org/10.1109/INFCOMW.2011.5928903

[Aldrich(2005)] John Aldrich. 2005. Fisher and Regression. *Statist. Sci.* 20, 4 (2005), 401 – 417. https://doi.org/10.1214/088342305000000331

[Breiman(2001)] Leo Breiman. 2001. Random forests - machine learning. https://link.springer.com/article/10.1023/A:1010933404324

[Corporation(2024)] Visual Crossing Corporation. 2024. Total weather data: Historical Weather Weather Forecast Data. https://www.visualcrossing.com/weather-data

[Departement of California(2024)] CalHHS Departement of California. 2024. Influenza surveillance - dataset - california health and human services open data portal. https://data.chhs.ca.gov/dataset/influenza-surveillance

[Eysenbach(2006)] Gunther Eysenbach. 2006. Infodemiology: Tracking flu-related searches on the web for syndromic surveillance. https://pubmed.ncbi.nlm.nih.gov/17238340/

[Fluview(2024)] CDC Fluview. 2024. National, regional, and state level outpatient illness and viral surveillance. https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

[Friedman(2001)] Jerome H. Friedman. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics* 29, 5 (2001), 1189–1232. http://www.jstor.org/stable/2699986

[Hedengren(2023)] John Hedengren. 2023. Data-Driven Engineering. https://apmonitor.com/dde/index.php/Main/AutoRegressive#:~:text=The%20ARX%20model%20combines%20both,and%20a%20random%20error%20term.

[MH;(2024)] Shih DH;Wu YH;Wu TW;Chang SC;Shih MH;. 2024. Infodemiology of influenza-like illness: Utilizing google trends' big data for epidemic surveillance. https://pubmed.ncbi.nlm.nih.gov/38610711/

[Pearson(1895)] Karl Pearson. 1895. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London* 58 (1895), 240–242. http://www.jstor.org/stable/115794

[Petermeissner(2018)] Peter Petermeissner. 2018. Wikipediatrend: A convenience R package for getting Wikipedia article access statistics (and more). https://github.com/petermeissner/wikipediatrend

[RE;(2013)] Dugas AF;Jalalpour M;Gel Y;Levin S;Torcaso F;Igusa T;Rothman RE;. 2013. Influenza forecasting with Google Flu trends. https://pubmed.ncbi.nlm.nih.gov/23457520/

[Rogan(2024)] Kathleen Rogan. 2024. Data project. https://github.com/Tiny-boot/Data_project

[US(2024)] FRED US. 2024. Federal Reserve Economic Data. https://fred.stlouisfed.org/release?rid=118

[Wiki page(2007)] Wikipedia Wiki page. 2007. Page view statistics for Wikimedia projects. https://dumps.wikimedia.org/other/pagecounts-raw/

[Wikitools(2024)] Wikipedia Wikitools. 2024. Page view Analysis. https://pageviews.wmcloud.org/?project=en.wikipedia.org&platform=all-access&agent=user&redirects=0&range=latest-20&pages=Cat%7CDog