COMP 551 - Project 1 Report

Analyzing COVID-19 Search Trends and Hospitalization

Group 28
Linda Cai (260720706), Jennie Chen (260745736), Zhengwen Fu (260856256)
October 22, 2020

Abstract

In this project, various machine learning algorithms and strategies are applied to analyze the search trends of symptoms and to predict the hospitalization cases based on their patterns. Two datasets respectively on the search trends of symptoms and the hospitalization cases were cleaned and merged for use. Visualization, clustering, and dimension reduction were performed on the search trends of symptoms to understand the feature data clearly. k-nearest neighbors algorithm and decision tree algorithm are applied to train the models for predicting the hospitalization cases based on the processed dataset. Their regression performances are compared and analyzed. The result shows that using decision tree on the whole dataset achieves the best regression performance for this dataset and is generally fast to train. The result also shows that the data from one region is less solid for making predictions for other regions. Finally, we concluded that the k-nearest neighbors algorithm is suitable for datasets with closely correlated data, whereas the decision tree algorithm excels for datasets of larger size.

1 Introduction

During the COVID-19 pandemic, the varying number of hospitalization cases concerns the general public. In this project, to investigate the link between the search trends of symptoms and the hospitalization numbers, we followed the three main steps below.

First, the two datasets for this project, the COVID-19 Search Trends symptoms dataset [1] and the COVID hospitalization cases dataset [2], collectively provide insights on the evolution of the search trend and corresponding hospitalization cases over time. Through reshaping, filtering [3], normalizing [4] and merging the datasets [5], the symptoms are able to convey distinctive patterns through a subtle correlation with region-based hospitalization.

Second, principle components analysis (PCA) is used to reduce the dimension of the search trends dataset and K-means is used as the clustering algorithm to classify the data to determine any possible relationships within the dataset.

Finally, two supervised learning algorithms are applied to the datasets, using symptom search trends as the features and hospitalization cases as the label. The two algorithms are the k-nearest neighbors algorithm (KNN) [6] and the decision tree algorithm [6]. Two validation strategies are applied to each of the two learning algorithm in order to study the regression performance: 5-fold cross-validation with region-based data splitting, and validation with time-based data splitting. The mean squared error (MSE) [7] is used to represent the performance, with a smaller value implying better performance. Furthermore, one extra prediction strategy, learning separate models for each region, is utilized to predict differently. Time-based validation is applied to this prediction strategy. In this project, the Scikit-learn library [8] is used to perform the learning algorithms and validation strategies.

2 Datasets

2.1 Pre-processing

The COVID-19 Search Trends symptoms dataset reflects on the volume of Google searches related to COVID-19 health symptoms across different US regions. The dataset given covers a broad spectrum of symptoms that may not be necessarily complete in terms of its presence in all the regions. In order to extract useful information

to convey a specific pattern related to the trends of searches, the pre-processing task is broken down into sub-tasks of data filtering and data normalization.

2.1.1 Data filtering

For data filtering, it makes sense to first clean symptoms that do not appear in the search in any given date or region; in other words, we remove features that contain no value in their given column in the dataset. We observe a reduction in the size of dataset after this step: $(608, 430) \Rightarrow (608, 127)$.

Although the rest of the symptoms certainly contains at least 1 non-empty entry, it does not make much sense to preserve such symptom as it has minimum effect on the overall algorithms. To examine the data trend accurately, we set a threshold for the set of symptoms so that each has sufficient data points to work with. We derived a table and a graphical chart to investigate the effect of a threshold on the size of dataset. As we increase the threshold of minimum non-empty entry for certain column, we see the following changes:

ratio	#columns	#symptom	% relative data loss
0	124	120	
0.1	105	101	-15.8333
0.2	81	77	-23.7624
0.3	19	15	-80.5195

Figure 1: Table:Threshold vs Data

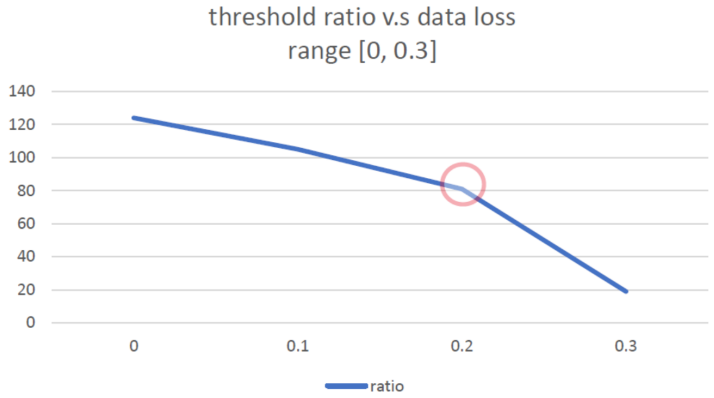


Figure 2: Graph:Threshold vs Data

To prevent losing too many symptoms for our dataset, we chose the optimal threshold value while preserving the symptoms that are significant, and that threshold value is 0.24.

2.1.2 Data normalization

As the dataset documentation denotes, the region-specific normalization factor renders it difficult to compare the numbers across different regions. In order to rid of the scaling factor, we applied Median normalization (MedN), which divides each value by its overall state symptom median and this will simultaneously preserve the search popularity over time.

On the other hand, the COVID hospitalization cases dataset aggregates information about the change in recorded hospitalization cases over time (daily), and the chosen feature to analyze is 'hospitalized new' column. The data filtering process for this dataset involves several steps: including reshape the dataframe, filter out the US regions data, truncate data to valid date-range, drop regions without any hospitalization data, merge daily rows into weekly rows and sum over the data values.

Finally we merged the two datasets into one single numpy array for subsequent tasks on visualization and supervised learning algorithms. The processed dataset consists of sample data for 11 regions ranging from 2020-03-09 to 2020-09-21 at the weekly resolution.

2.2 Visualization of the popularity of symptoms across regions and over time

To visualize the data, we used two different methods to facilitate the comparison of the search trends for different symptoms across all regions and over time. For both of these methods, only the five most popular symptoms were selected and visualized for simplicity. The choice of these five symptoms was made based on the number of non-zero instances for each symptom.

2.2.1 Comparison across regions

To visualize the data at different time points, we use ipywidgets library's slider functionality to visualize a heatmap of the regions and the five most popular symptoms through time. A sample of the heatmaps, generated for the first three weeks of the time range, is shown in Appendix A. It should be noted that this visualization scheme was inspired by the official visualizer for the COVID-19 Search Trends symptoms dataset [9].

Through this visualization, we can make a few observations. First, we see that the regions Hawaii, Maine and New Mexico have relatively high search frequency ratios to the median for all symptoms. We can also observe that the search trend for aphonia in the region of New Hampshire consistently remains above 2x of the median for all time. Lastly, we can observe that there aren't any trends within each symptom for all regions. In other words, at any time instance, there isn't a symptom whose search trend is consistently above or below median for all regions.

2.2.2 Comparison over time

To complement the heatmaps and to visualize trends over time, we use histograms to plot the normalized search frequency data, averaged across all regions, over the entire time period. Histogram bins are colored to provide visual indication of their value with respect to the median.

The graphs, generated for each of the five most popular symptoms, are shown in Appendix A. By observing the histograms, we can see that during the first 4-5 weeks of our time range, the search frequencies of all five symptoms are higher than the median.

2.3 Pre-processing for supervised learning

NaN values are replaced with 0's, because in this dataset no observed data for such symptom often means this symptom does not exist there indeed. The dataset are also respectively sorted by regions and by time for data splitting in supervised learning.

3 Results

3.1 Data dimensionality reduction

To reduce the data dimensionality of the search trends dataset, PCA is performed with the help of the sklearn library. The original dimension was 61, which corresponds to the number of symptoms in the dataset. To determine the number of principle components (PCs) to keep, we plot the cumulative variance explained versus the number of PCs, as shown in Appendix B.1. By setting the cumulative variance threshold to 95%, we determine the optimal number of PCs to be 10. This means that by using the first 10 PCs, we can explain at least 95% of the variance in the original data, effectively reducing the dimension from 61 to 10. Fig. 3 shows a plot of the first two PCs of the search trends dataset that has been reduced to 10 dimensions.

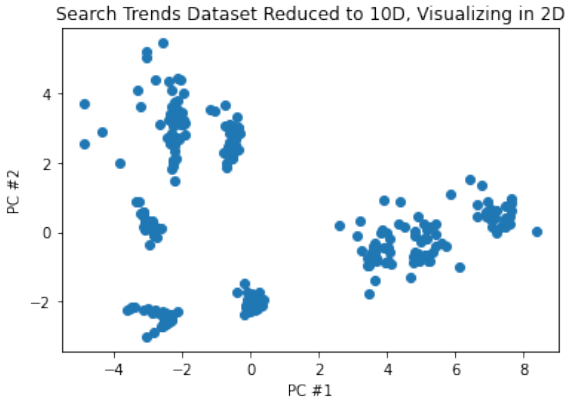


Figure 3: Visualizing the first 2 PCs of the search trends data reduced using PCA

3.2 Data clustering

To cluster the data into groups with similar characteristics, we use K-means, implemented by sklearn. This is performed on both the reduced (10 dimensions) and original (61 dimensions) data.

To determine the optimal number of clusters (K), we first plot the inertia of the data, which is the sum of squared distances of each point to its closest cluster center [10], at different values of K (see Appendix B.2). From the plot, we use the elbow method to visually determine the optimal number of clusters [11], which is shown to be K = 6 for both the reduced and original data.

Then, K-means is performed for both sets of data and the resulting clusters, plotted for the first two dimensions (PC1 and PC2), are shown in Fig. 4.

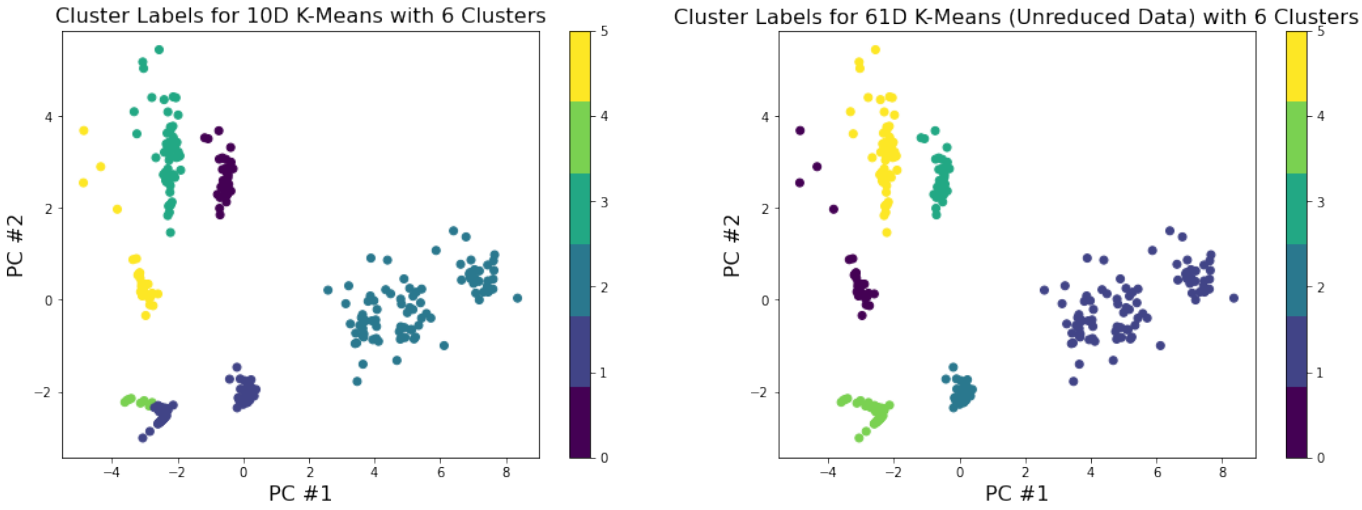


Figure 4: K-means clustering on reduced and original data for K = 6 clusters

From Fig. 4, we can see that the only visible difference between the clusters in both plots is the split between clusters 1 and 4 in the reduced data, and between clusters 2 and 4 in the original data. Returning to the inertia calculations, obtained at K = 6, we note a value of 1522.2551138411916 for the reduced data, and a value of 1938.5104655936402 for the original data. The lower value of inertia in the reduced data's case indicates that for the same K, the datapoints are overall closer to their cluster centers. Since inertia measures the interal coherence of the clusters, this indicates that by using PCA to reduce the dimensionality of the data, the clusters are better defined and the data in each cluster are more correlated to each other.

3.3 Hyperparameter determination for learning algorithms

The plots for the performance of the algorithms vs. the most important hyperparameter during training models are attached to Appendix B.3. The plots are for finding the best value of the hyperparameter that can represent the performance of the algorithm. For KNN models, K is chosen to be the local minimum near the $\sqrt{N}$ (N is the number of samples), which is best balance of performance and time complexity. For decision tree models, the max_depth is chosen to be the global minimum, which is always a very small value.

3.4 Comparison of regression performances

The time-based validation error of the model trained on the whole dataset using decision tree is the lowest among all prediction and validation strategies, as shown in Table 1. Moreover, the training time of this model is the shortest among all prediction and validation strategies, as shown in Table 2.

The region-based cross-validation errors on both KNN and decsion tree are much higher than the time-based validation errors.

Both with time-based validation, the performance of KNN on separated regional data is better than that on the whole dataset, as shown in Table 1. However, the performance of decision tree on separate regional data is worse than that on the whole dataset, as shown in Table 1.

Table 1: Validation error (MSE) of learning algorithms with different validation strategy

	Region-based	Time-based	Time-based (regional models)
KNN	8.405×10^3	2.799×10^3	2.596×10^3
Decision tree	6.379×10^3	2.067×10^3	2.789×10^3

Table 2: Training time of learning algorithms with different validation strategy (the specific values may vary during different runs)

	Region-based (s)	Time-based (s)	Time-based (regional models) (s)
KNN	13.631×10^{-3}	3.022×10^{-3}	6.013×10^{-3}
Decision tree	8.995×10^{-3}	1.969×10^{-3}	5.670×10^{-3}

4 Discussion and Conclusion

Reducing the data dimensionality using PCA effectively improved the clustering of the data using K-means. This can be seen by the reduction in the inertia, a measurement of the internal coherence of the clusters, after performing dimensionality reduction and k-means clustering.

Since the performance and the training time of decision tree on the whole dataset are both the best in all prediction and validation strategies, it can be concluded that using decision tree on the whole dataset achieves the best regression performance for this dataset and is generally fast to train.

The errors in regional-based validations are really high. This is actually expected, because the sample data from one region is relatively less solid for making predictions for other regions.

After splitting the whole dataset into datasets of different regions and training on each of them separately, it is expected that the data in each regional dataset are more closely correlated while each dataset is of relatively smaller size. Since KNN has a better performance and decision tree has a worse performance in such condition, it can be concluded that KNN is good for closely correlated data and is less sensitive to smaller dataset size, whereas decision tree is good for datasets of large size and is less sensitive to the closer correlation of data.

For future work, it would be interesting to include the COVID-19 Search Trends symptoms dataset at the daily resolution [2]. According to the description on how the data was collected, the daily time series contains more accurate data and is populated first. If the daily data cannot be accurately provided, then weekly data is generated instead. In other words, the two datasets are complimentary to each other. Therefore, including the daily dataset would provide a more global insight on the actual search trends.

5 Statement of Contributions

Linda Cai contributed to the data preprocessing and report writing.

Jennie Chen contributed to the data visualization and clustering, and report writing.

Zhengwen Fu contributed to the supervised learning and report writing.

References

[1] "Covid-19 search trends symptoms dataset." [Online]. Available: https://github.com/google-research/open-covid-19-data/blob/master/data/exports/search_trends_symptoms_dataset/README.md 1

[2] "Open covid-19 data." [Online]. Available: https://github.com/google-research/open-covid-19-data 1, 5

[3] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 856–863, 2003. 1

[4] K. Fundel, J. Haag, P. Gebhard, R. Zimmer, and T. Aigner, "Normalization strategies for mrna expression data in cartilage research," 2008. 1

[5] V. Yordanov, "Data science with python: Intro to loading, subsetting, and filtering data with pandas," 2018. 1

[6] K. Murphy, *Machine Learning: A Probabilistic Perspective*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2012. [Online]. Available: https://books.google.ca/books?id=RC43AgAAQBAJ 1

[7] D. Wackerly, W. Mendenhall, and R. Scheaffer, *Mathematical Statistics with Applications*. Cengage Learning, 2014. [Online]. Available: https://books.google.ca/books?id=lTgGAAAAQBAJ 1

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. 1

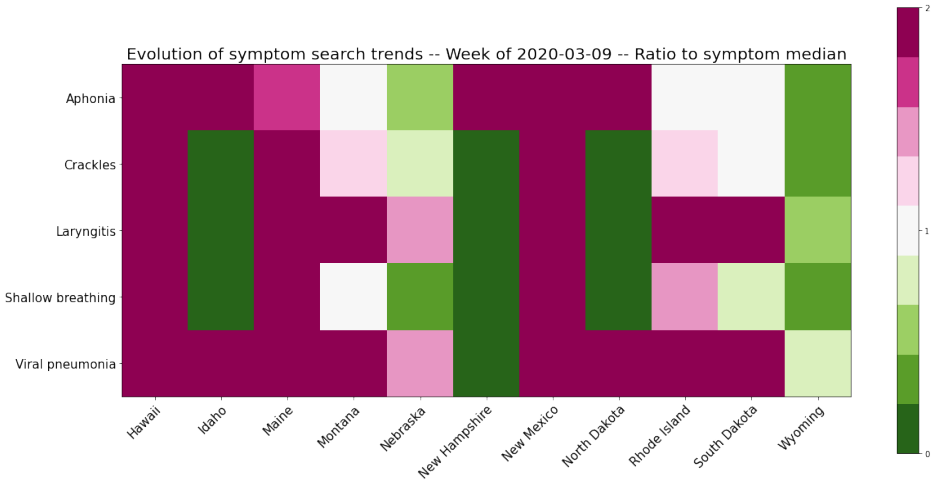
[9] "Explore covid-19 symptoms search trends." [Online]. Available: https://pair-code.github.io/covid19_symptom_dataset/?date=2020-09-07 3

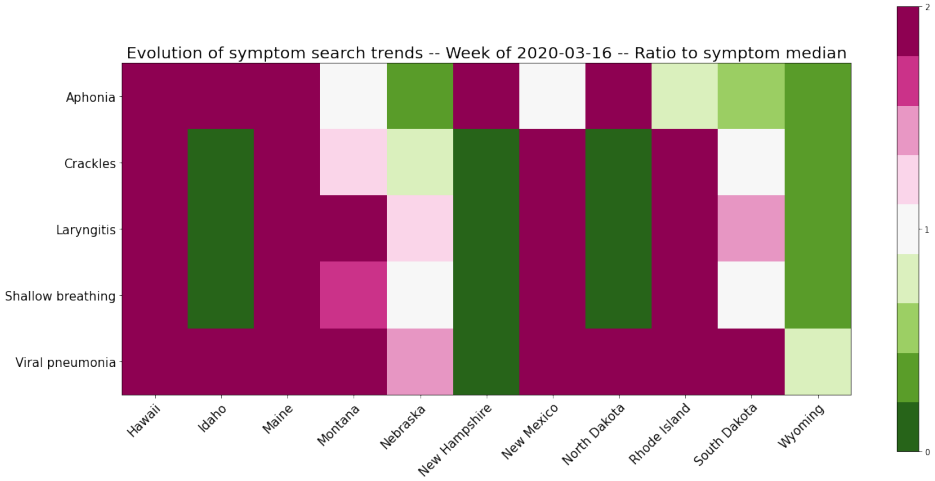
[10] "sklearn.cluster.kmeans." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html 4

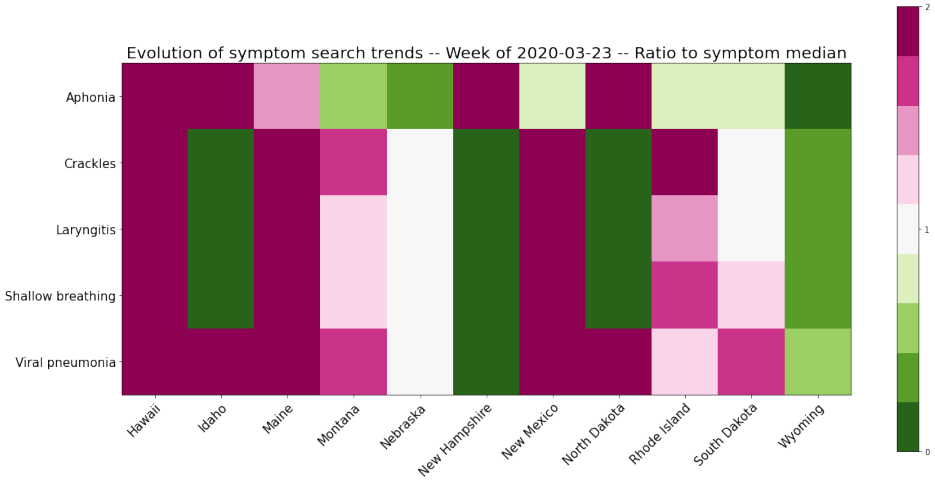
[11] S. J. Franklin, "Elbow method of k-means clustering algorithm," Nov 2019. [Online]. Available: https://medium.com/analytics-vidhya/elbow-method-of-k-means-clustering-algorithm-a0c916adc540 4

A Visualizations

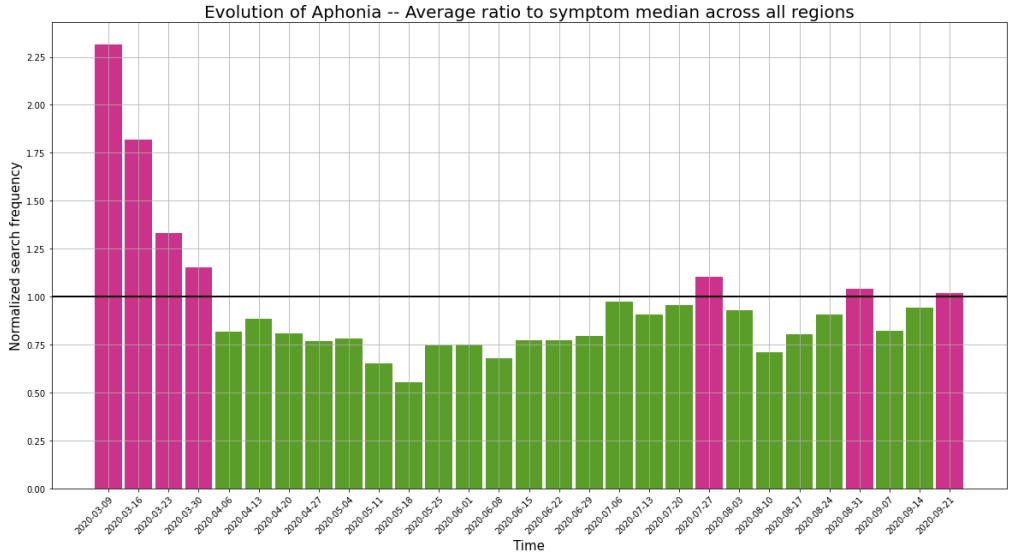
The search trend heatmap generated for the first three weeks (subsequent weeks are shown in the jupyter notebook):

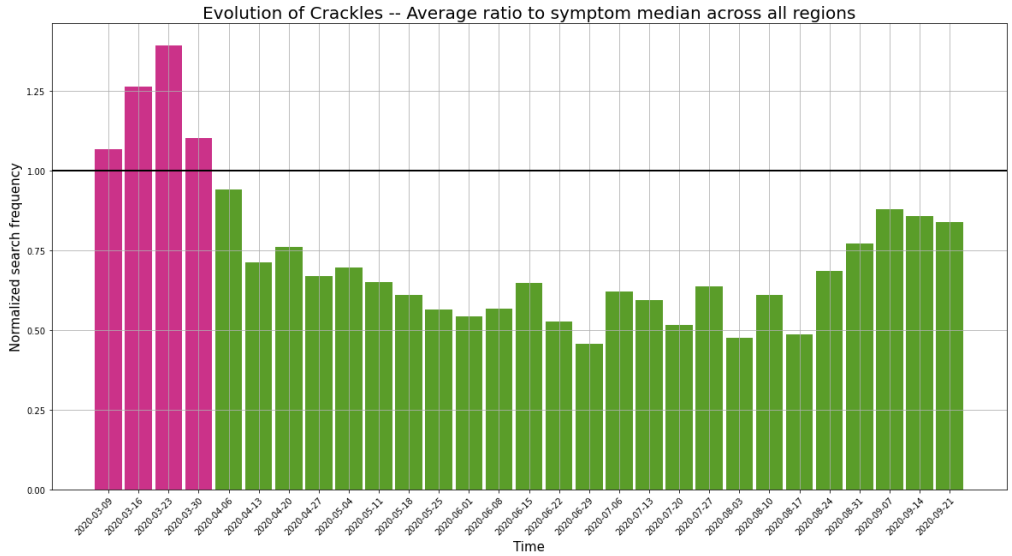


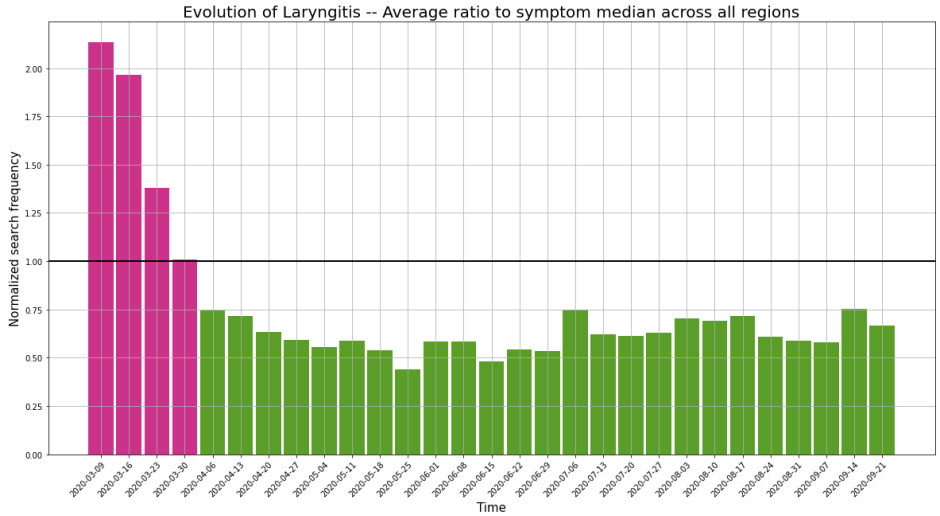


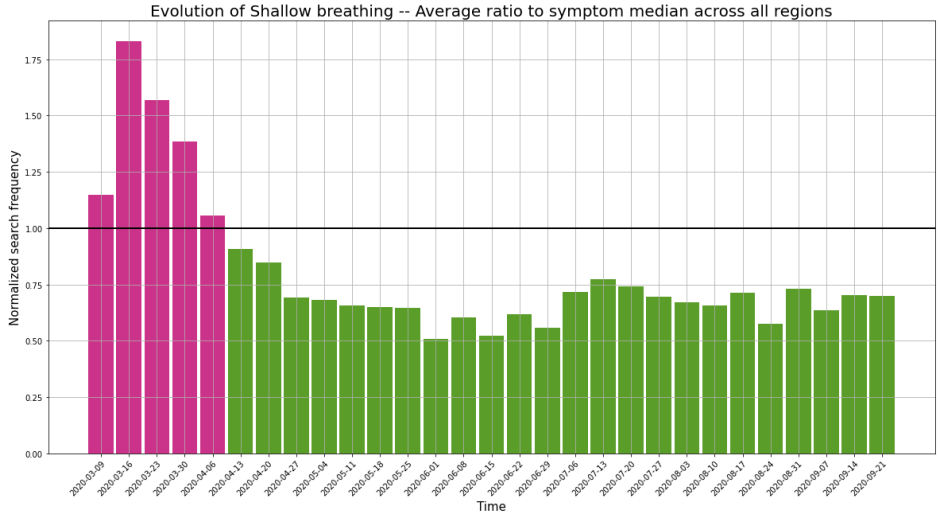


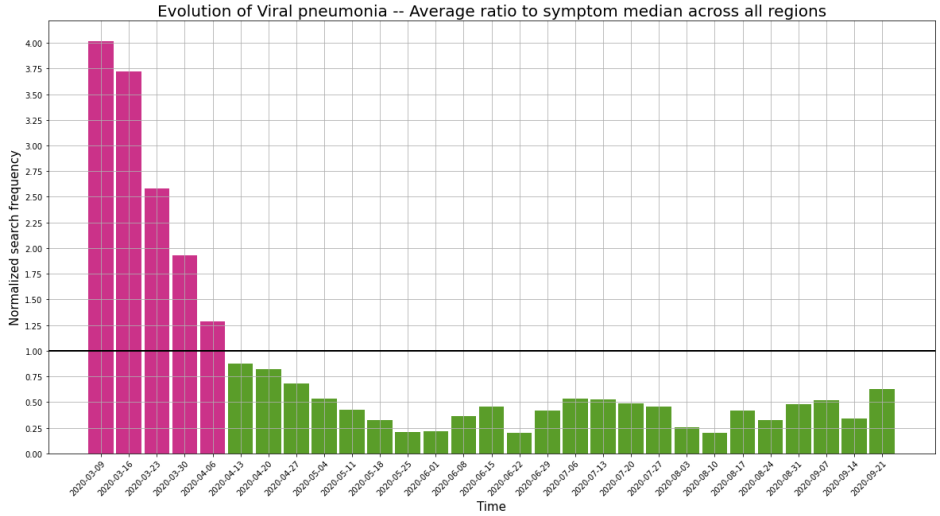
The histograms of the five most popular symptoms' search trends over the entire time range:








Evolution of Shallow breathing -- Average ratio to symptom median across all regions
Normalized search frequency
Time


Evolution of Viral pneumonia -- Average ratio to symptom median across all regions
Normalized search frequency
Time

B Plots for Hyperparameter Tuning

B.1 Hyperparameters in Dimension Reduction

To tune the number of PCs to be used in PCA, the following plot is generated as reference:

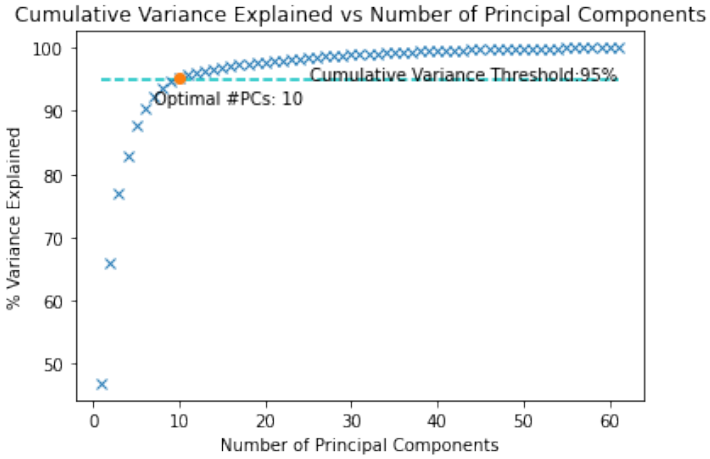


Figure 5: Plot of cumulative variance for PCA

B.2 Hyperparameters in Data Clustering

To tune the number of clusters (K) to be used in the K-means clustering algorithm, the following plots are generated to visualize the optimal number of clusters for both the reduced and unreduced (original) datasets:

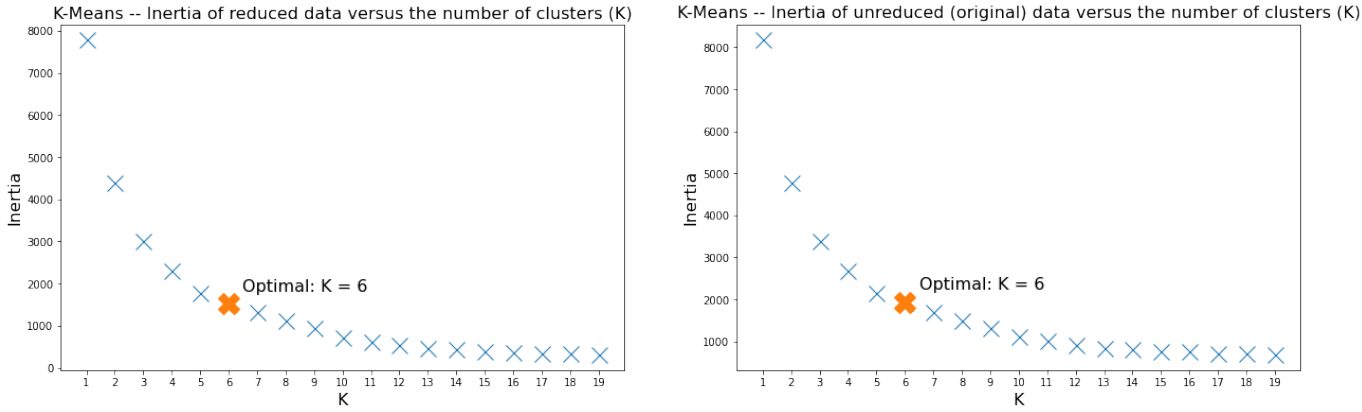


Figure 6: Plot of the inertia of the data for K = 1 to 20

B.3 Hyperparameters in Learning Algorithms

Plots for determining hyperparameters (first row: time-based validation; second row: region-based validation; third row: time-based validation for regional models):