COMP 551 Fall 2020

Mini Project 1: Analyzing COVID-19 Search Trends and Hospitalization
Written Report

Group 17
Haowei Qiu (260762269) Liang Zhao (260781081) Ningchen Ma (260784506)

1 Abstract

In this project, we first clean and merge two raw datasets (Search trends dataset and COVID hospitalization cases dataset) together. We then tried to visualize the merged search trends (higher-dimensional) dataset on a lower dimensional space by using the PCA method. We then did the potential group-classification on this data set by using the K-Means method and compare its performance on both the reduced and raw dataset.
Moreover, we investigated the performance of two regression models, namely k-nearest neighbours and decision trees, on predicting COVID-19 hospitalization cases from related symptoms search. We found that the k-nearest neighbour (KNN) regression approach achieved lower mean squared error than decision trees and was slightly faster to train.

2 Introduction

During this project, we need to first extract useful information from 2 data sets and then combine them together by appending hospitalization column to dataset 1. The merged file is in weekly basis and is started from 2020-03-16 to 2020-09-21. (Those two raw datasets we used throughout our analysis was updated until **10.05**.)
After that, we first visualized the search trends dataset in a lower dimensional space (2d or 3d space). To do this, we introduced the PCA method, which is an analytical method to reduce the dimension of dataset while gives little distortion when recover the data back. We then explored the grouping of our data by using the K-Means method. For most of time, using only 2 to 3 PCs can give us up to 90% variance of the dataset. Which indicates the powerfulness of the PCA method in dimension reduction.
In the last task, we compare performances of KNN and decision trees, with 5-fold Cross Validation, to predict the hospitalization cases given the search trends data. We used 2 types of data splitting strategy for 5-fold cross validation: split by region and split by time.

3 Datasets

We filtered out seldom-used data by setting a threshold so that symptoms with more than 95 percent data empty will be removed. We set the threshold low because we try to keep as much useful data as we can, but the downside is that this will create noises for future analysis. We used pandas function 'resample' to convert the data to weekly basis from 2020-03-16 to 202-09-21 and the merged data is stored in merged.csv.
For our exploratory analysis, in order to get an intuitive understanding to the evolution

trend for some popular symptoms, we introduced the line graph, because line graph is quite informative when dealing with tracking changes over short or long periods of time. They are also useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded.

Since visualizing a data set with more than 3 dimensions would be difficult or even impossible. Thus we need to find a way to represent our data more understandable and extract useful information as much as possible. That where the PCA comes to help!

During the process of doing PCA, we first found the desired number of principle components for our data, we then project the data set on those PCs to get the compressed data with dimensions equal to the number of PCs we picked.

As for the K-Means, we simply do the classification analysis on the reduced data from PCA and the raw data to discover potential grouping by giving some hyper-parameter, which somewhat depends on our intuition and knowledge to the data itself.

4 Results

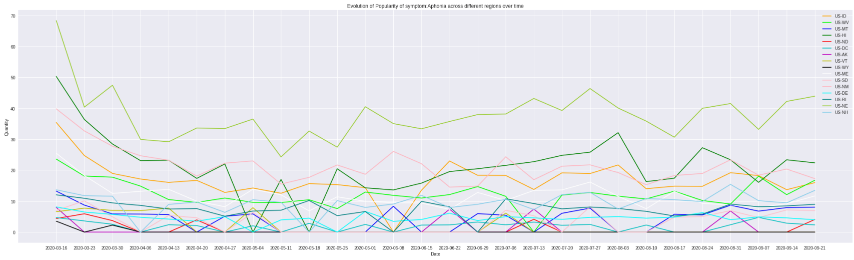
Part 2: Visualizing Data

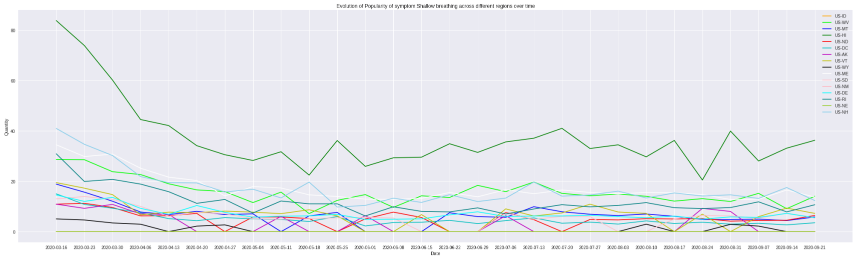
2.1: Evolution cross region over time

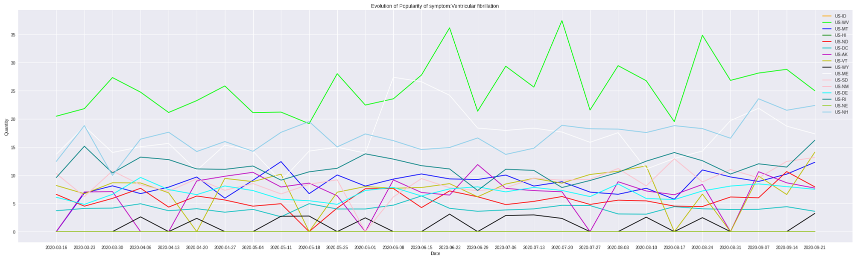
Here,we chose 3 most popular symptoms to do the analysis.

We choose those with the least number of non-zero entries. That is, symptoms with the most frequent record are considered to be "popular".

We used line graphs to represent the evolution of symptoms across regions over the time.

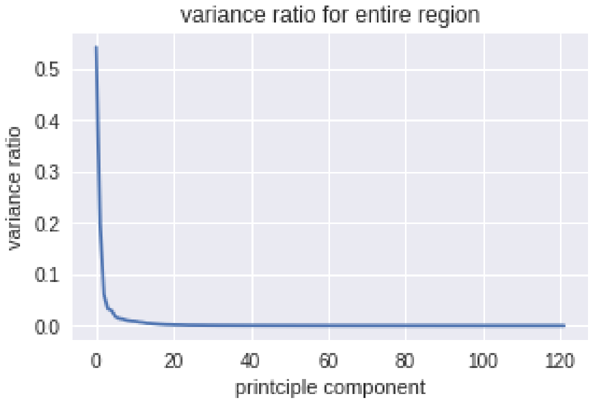


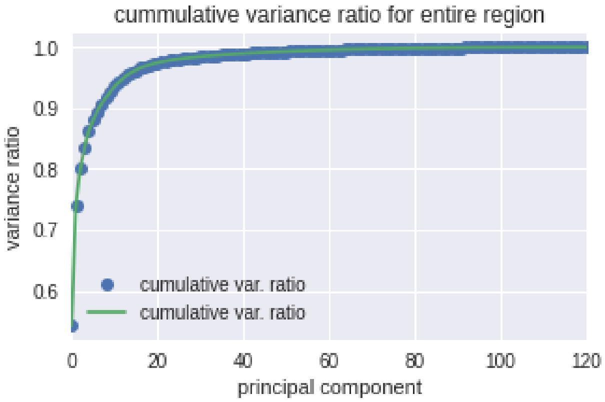




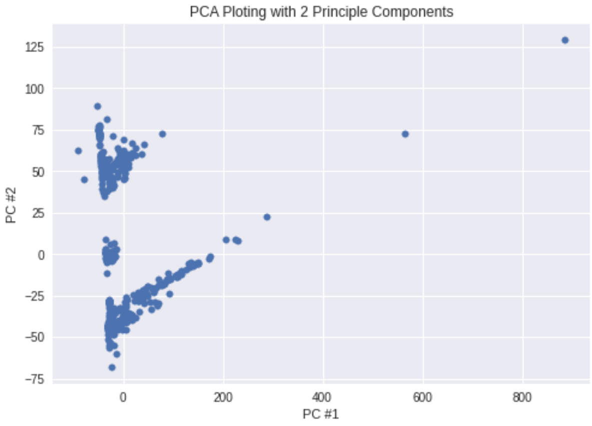
Note that all of our data has been normalized before the plotting, since the dataset has a region-specific normalization factor that makes it difficult for us to compare the numbers across different regions. The method we used to normalized the data is to find the median for each symptom search within a region, then find the median of the symptom search medians for each region. In each region, we divide the symptom search by the median found for the specific region to get the normalized data.

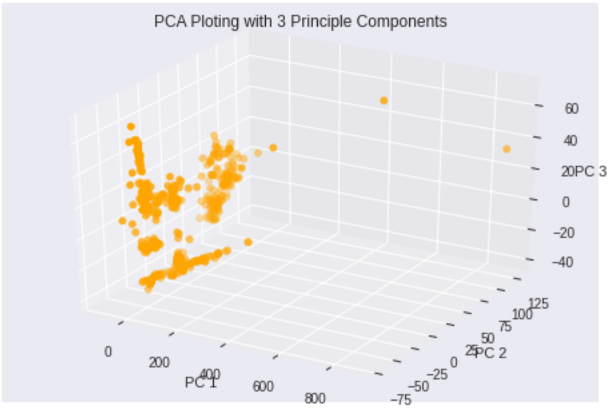
2.2: PCA





To do this analyzing, I treated each time point in the raw data set as an independent data point. Here a use 2 PCs would give us more than 70% cumulative variance, and a 3-PCs PCA can give a cumulative variance for more than 80%! Therefore, we consider both using 2 or 3 PCs as faithful PCAs. (Visualizing a PCA with more than 4 PCs would be difficult!)



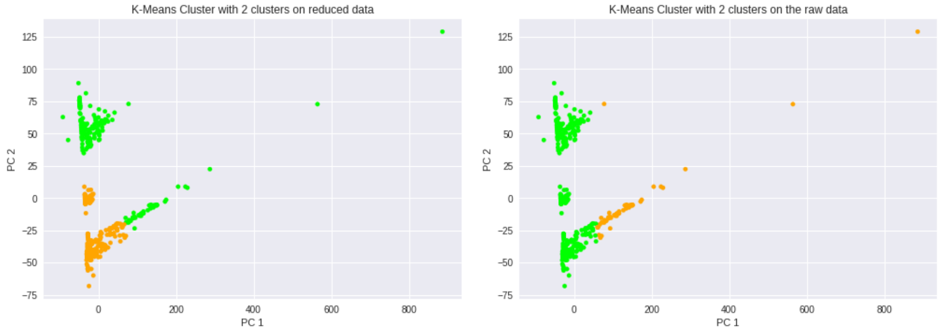


2.3: K-Means Cluster

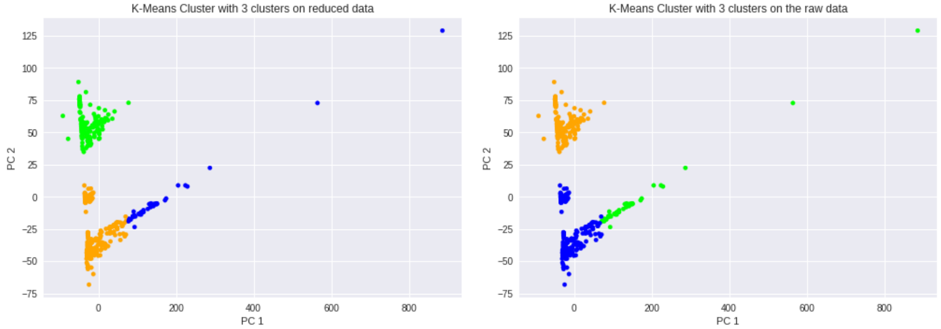
For the K-Means classification, we will only use 2-d PCA graph to do the analysis. Because it is easier to visualize ”groups” on a 2-d plane and such PCA graph is also informative enough for our purpose of usage.

In the following graphs, each color represents a unique group classified by the K-Means.

When k=2:



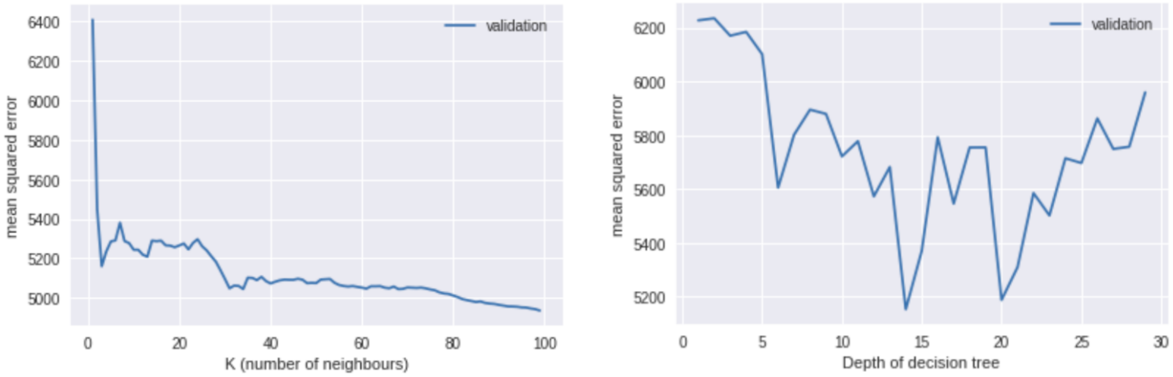
When k=3:



Here we can see that the classification for both raw and PCA data set are quite similar, but with some group "swapping", which in general will not impact our analytical conclusion. We choose hyper-parameter k as 2 and 3, since our data is pretty compact on the PCA graph. Hence intuitively, there should be no more than 3 groups in total.

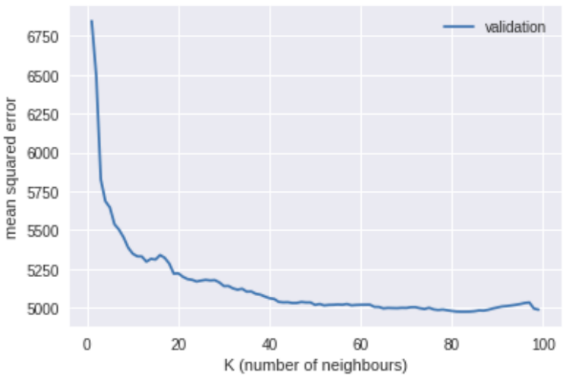
3. A comparison of regression performance (mean squared error) between KNN and decision trees on the cross-validation schemes

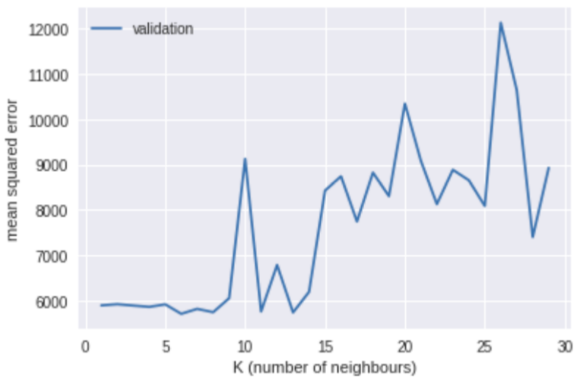
3.1 Data split based on Regions



The figures above illustrates the results of KNN (Left) and Decision Tree (Right) with cross-validation schemes split by regions. We observed that KNN will have lower mean squared error when K value increases. For Decision Tree, when the depth increases, the mean squared error will first decrease then increase, which might be caused by overfitting. In general, when data set is split by regions, KNN will have better performance, compared to Decision Tree.

3.2 Data split based on Time





The figures above illustrates the results of KNN (Left) and Decision Tree (Right) with cross-validation schemes split by time. We observed that KNN will have lower mean squared error when K value increases. For Decision Tree, when the depth increases, the mean squared error first remains the same then increases, which might be caused by overfitting. In general, when dataset is split by time, KNN will also have better performance, compared to Decision Tree.

Discussion and Conclusion

The line graph showed us the evolution trend across regions over time, however, we realized that leaving too many ”lines” (independent data sets) on a single graph could make it somewhat hard to compare and interpret. Therefore, we may choose to represent such content by using other forms of graph, such as heat-map to get a better intuitive understanding to the desired data.

The PCA method was used to successfully reduce the dataset dimension The K-Means clustering performs equally well on both the reduced and raw dataset. However, since it is unsupervised method, we have no way to measure the accuracy whatsoever.

In general, KNN will have better performance, compared to Decision Tree both when data is split by regions and when data is split by time. The KNN performance will improve when K increases. Increasing Decision Tree depth, will first improve Decision Tree performance, but will get worse when depth value is too high, which is caused by over fitting.

We found that the results supervised training also depends on the result of dataset cleaning. Keeping features with insufficient amount of record is as to leave a lot of noise and outliers in the dataset, which could impact the precision of the following data analysis. Therefore, for future investigation, we will focus on improving our dataset cleaning strategy.

Statement of Contributions

The code implementation was assigned to each group member based on tasks. Specifically, Liang Zhao (ID: 260781081) contributed to Task 1, Ningchen Ma (ID: 260784506) contributed to Task 2, and Haowei Qiu (ID: 260762269) contributed to Task 3. All 3 group memebers participated in the mini project report writing.