COMP 551 - Mini-Project 1

Bouchard, William
Beaupré, Frédéric

October 21, 2020

Abstract

In this project we investigate the performance of two supervised learning frameworks, namely K-nearest neighbors (KNN) and decision trees (DT), on predicting COVID-19 hospitalization cases from related symptoms search trends in the US. We found that KNN performs better than DT on the datasets used even though the mean squared error is quite high for both models due to noise in the datasets; noise which we tried to reduce by using only symptoms heavily related to COVID-19, and by removing outliers in our predictions. In addition, while both models had decent training time, we noticed KNN was slightly faster than DT although we did not precisely measure the difference in time. We also explored the effects of dimensionality reduction on the outcomes and accuracy of the K-means method on our unlabeled dataset, i.e., the search trends dataset. We evaluate and compare models through hyper-parameter tuning, namely the choosing of K in K-means and KNN (for instance, via the elbow method), as well as the choice of dimensions to reduce to in principal component analysis.

1 Introduction

We studied the performance of KNN and DT at predicting COVID-19 hospitalization cases per US region per week based on the related symptoms search trends dataset provided by Google Research [1]. Google Research provides daily and weekly search trends for various symptoms across US regions. We used the daily dataset as it contained more information, and only then converted it to weekly resolution. The data between regions cannot be compared, as per the documentation: "In a single region, you can compare the relative popularity of two (or more) symptoms (at the same time resolution) over any time interval. However, you should not compare the values of symptom popularity across regions or time resolutions [...]" [1]. We discuss how to solve this problem in the *Datasets* section. The other dataset is the COVID hospitalization cases dataset, also provided by Google Research [2]. More specifically, we used the new hospitalized cases per region across time as our targets.

We found that KNN performs better than DT, and was also faster to train. The detailed results (mean squared error) of both frameworks are described in the *Results* section. In terms of the visualization of our data, when fitting a K-means model to it, we found that a number of clusters $K = 5$ is the value which gave the clearest and most meaningful clusters. Moreover, in performing dimensionality reduction (to 2D and 3D) on our data using principal component analysis, we can clearly see that there is a group of data points significantly detached from the rest, which could reduce the accuracy of our predictions, in the specific cases of cross-validation schemes where those detached points find themselves in the validation set.

2 Datasets[1]

The first dataset consists of normalized popularities of symptom searches on Google per day for different regions of the United States. Throughout this report we will call this data the 'search trends'. The second dataset includes information about COVID-19 cases, deaths, tests, and other features per day for different regions across the globe. This dataset we will call 'covid cases'.

The first step in our pre-processing process was to keep only the US-regions from the covid cases that also appear in the search trends. We then convert the two datasets, which were originally in a daily resolution, to a weekly resolution. For the search trends dataset, we then dropped the symptoms that had over 30% NaN values. To handle the remaining NaNs, we elected to replace them by zeros. Note that other valid approaches would have been to replace them with the median or mean of the corresponding column for that NaN value's US region. We went with the fill-by-zero approach because it is difficult to disambiguate missing data from actual zero data.

After reading the documentation, we also noticed that the search trends could not be compared between regions because the region-specific scaling render comparisons between them meaningless. To solve this problem, we divided each entry in our dataset by the median of all entries that have the same region code. By doing so, we normalize the data to a different metric which we can then compare across regions. This idea of normalizing using the median was derived from a similar idea used by Google on its *Explore COVID-19 Symptoms Search Trends* web page. It is stated that "to make trends in Google searches comparable between counties, we divided the normalized search volume for a given symptom in a given county during a given week by the median search volume for that county-symptom pair" [3] . We then apply a standard scaler specific to each group of regions, so that each group's values have a mean of zero and unit standard deviation.

Analysis of the covid cases then showed that a lot of states had a large number of zero values. We remove these states from the dataset with an arbitrary threshold of about 33%. For most regions we removed, the amount of missing information was clearly above threshold, except for NY and NJ. We decided to remove New Jersey but keep New York, as the latter had slightly more valid information. We finally merged the two datasets based on region code-date pairs.

[1]The versions of the datasets used are those of 2020-10-20

3 Results

3.1 Search Trends Data Visualization

We started visualizing the data by plotting the search trends popularity of various symptoms for a given region. Below are the resulting plots for notably popular symptoms (i.e.: Anxiety, Infection, Pain, Fever, Cold, Cough) against time for Colorado.

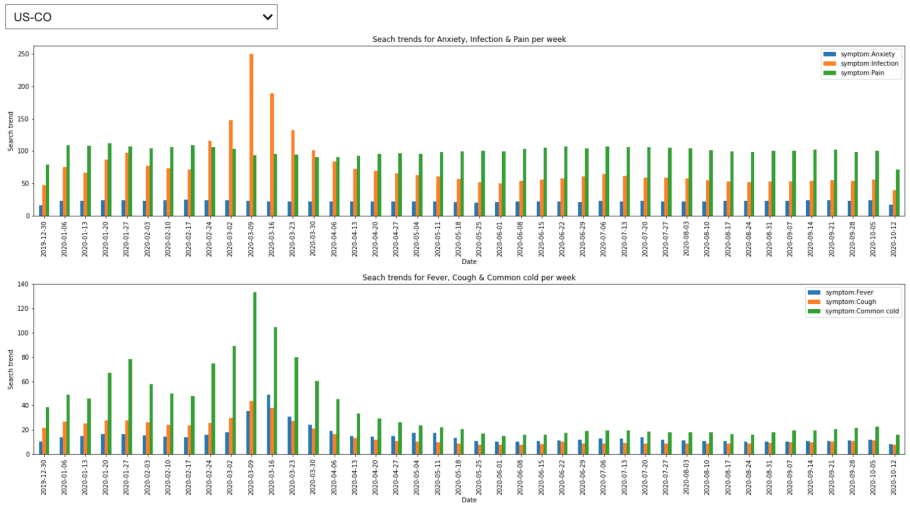


Figure 1: Visualization of the search trends across time for Colorado.

More specifically, these histograms depict the normalized search volume per week for the symptoms listed above. Note that for most of the regions, if not all of them, we can clearly identify peaks in the popularity of Infection and Common Cold search trends around March 2020. While this may be related to the outbreak of COVID-19, we must not be too quick to draw conclusions as these dates also match the yearly flu season in America.

3.2 PCA-reduced data and hyper-parameter tuning

We then proceeded to perform dimensionality reduction on our dataset through principal component analysis. Initially, the data consisted of 422 symptoms, hence 422 dimensions. For visualization, we reduced this to 2 and 3 dimensions, which resulted in the graphs below, where we can clearly see the directions of variance in the principal components' directions.

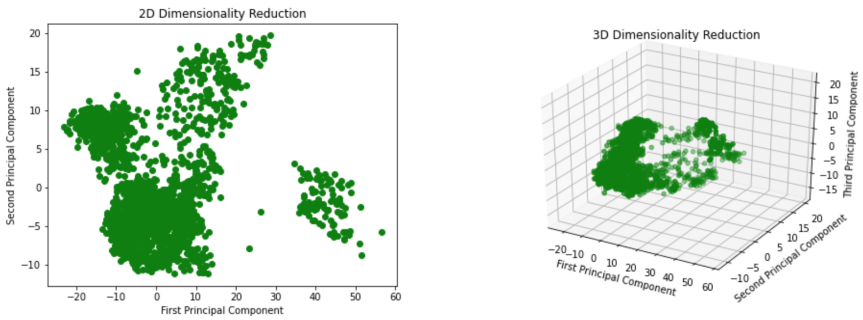


Figure 2: Visualization of the search trends data in 2 and 3 dimensions.

We can observe in Figure 2 that there is a small concentration of points that is detached from the rest in 2D, and that a similar separation occurs in 3D. Indeed, there appears to be a large concentration of points in the lower values of the first and second principal components In PCA, the number of dimensions D is a hyperparameter which we can fine-tune in order to get the best possible model for our dataset. Common practices include choosing D such that we ensure that we retain 90% to 99% of the dataset's variance. In our case, this amounted to 75 dimensions for 90% variance, 160 dimensions for 95% variance, and 300 dimensions for 99% variance.

We investigated further by looping through all the number of dimensions we could reduce to using PCA (i.e: $D \in [1, 421]$) and calculate the variance and reconstruction loss for each iteration of the loop. We then considered the value of D where going further to $D + 1$ offered minimal gain in variance ($< 1\%$), which was $D = 7$. Finally, we also considered values of D from applying the elbow method on both the variance plot and the reconstruction loss plot, which both gave 30, as represented by the red markers in the plots below.

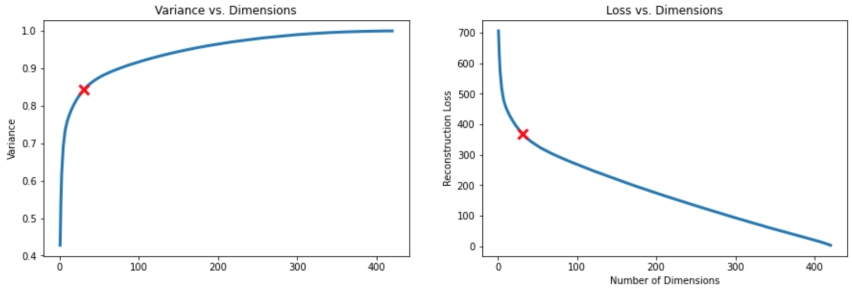


Figure 3: Elbow method for variance and reconstruction loss vs. number of dimensions.

To summarize our PCA hyper-parameter tuning analysis, we have explained how we explored different approaches in selecting the value of D which would give the best fit to our model. The values of D obtained from the different methods described above are:

$$D = 7, 30, 75, 160, 300$$

3.3 K-Means

We used K-means to evaluate possible groups in our dataset, where we want to partition our data into K clusters which minimize the sum of distances to the cluster mean/center. For the choice of the hyper-parameter K, we loop through numbers of clusters in the range (1, 15) and fit our dataset to a K-means model with that number of clusters. We plot the sum of squared distances of samples to their closest cluster center for each of the models and choose the best one using the elbow method. In our case, this results in a best-value $K = 5$, with 4 another, albeit not as promising candidate.

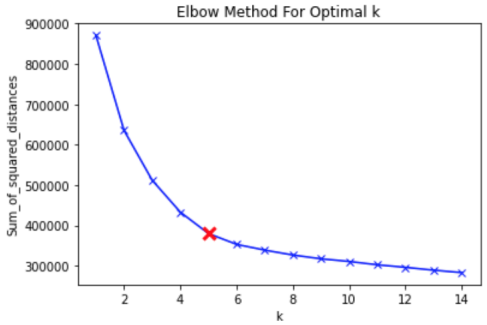


Figure 4: Elbow method for best choice of number of clusters.

To visualize the result of our K-means models for $K = 4, 5$, we plot the normalized and standardized raw data as well the PCA-reduced data with their cluster memberships in the graphs shown below:

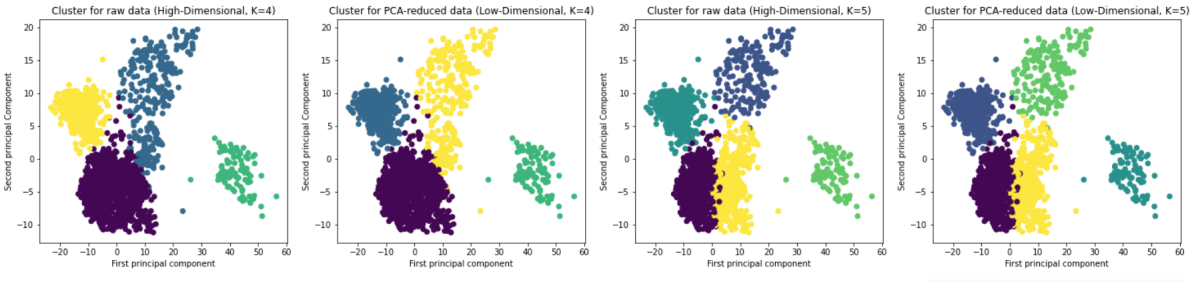


Figure 5: Clustering for $K = 4, 5$ of the raw and PCA-reduced search trends data.

We can see that the regions defined by the clusters for raw data and PCA-reduced data are extremely similar. However, the cluster memberships change for about 50% of the data points. One interesting observation is that the bottom region (in purple for $K = 4$, purple and yellow for $K = 5$) shows consistent cluster memberships for both raw and PCA-reduced data. Moreover, the isolated region on the right has its cluster membership change from raw data to PCA-reduced data for $K = 5$, but remains the same for $K = 4$.

Note that we also constructed plots of the cluster memberships for $K = 4, 5$ in 3D. Those can be found in *Appendix A*.

3.4 KNN and DT Regression

We performed KNN and DT regression using two different train-validation split strategies, namely region-based and time-based. For the former, we perform a 5-fold cross-validation in which 80% of the regions are used for training and the rest for validation. We first ran our KNN model for values of $K \in [2, 100]$ and computed the average mean squared error(MSE) of our predictions in each iteration, producing the figure below:

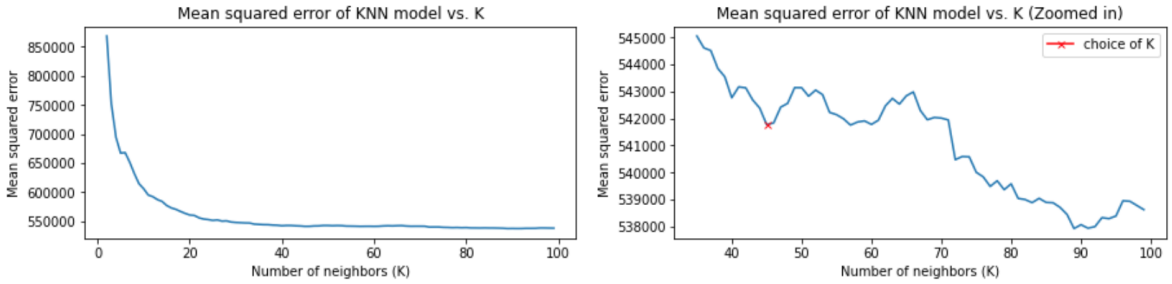


Figure 6: Average MSE of KNN model vs. K.

From studying these graphs we made the decision of choosing $K = 48$ as it appears to be the value with the best trade-off between accuracy and computational cost, and is a clear local minimum. After this fine-tuning of K, we obtain an average MSE of:

$$KNN : 542\,555 \quad ; \quad DT : 1\,425\,265$$

The other train-validation split strategy we used was time-based. All values after 2020-08-10 were put in the validation set and the rest in the training set. We repeated the method described above to find the best trade-off value of K for our model and obtained $K = 90$. This splitting strategy gave the following average

MSEs:

$$KNN : 198\ 481 \quad ; \quad DT : 13\ 553\ 526$$

We can then clearly see that for both cross-validation scheme, KNN performs way better than DT.

4 Discussion and Conclusion

While KNN performed better than DT, we still have a large MSE for both models. This is most likely due to a large amount of noise in the datasets. Even though we had already tried to reduce the noise by keeping only symptoms with enough data (70%), we tried to reduce it even further by experimenting with two additional preprocessing steps.

To begin with, we searched online and consulted a microbiologist-infectiologist expert [4] for the symptoms most heavily related to COVID-19 and kept only those in the dataset. The list of all these symptoms can be found in *Appendix B*. This proved to be an unfruitful method as our average MSE errors barely dropped for KNN and even increased for DT:

$$KNN : 535\ 121 \quad ; \quad DT : 2\ 719\ 300$$

Note that we used the region-based train-validation split for these results. This lack of increase in accuracy clearly shows the large amount of noise in the search trends. After careful analysis of the discrepancies between predictions and ground truth values, we noticed that the large MSE was probably caused by several outliers in our predictions. We aimed to validate this by removing said outliers and recalculating the MSE. As we expected, the MSE drastically dropped for both regression methods:

$$KNN : 39\ 441 \quad ; \quad DT : 55\ 783$$

In sum, KNN appears to be a better regression model for predicting new hospitalized COVID-19 cases from the search trends. However, the dataset is very noisy, which impacts the precision of our predictions. We aimed to reduce this impact by reducing noise through the two methods described above, one of which was fruitful. Moreover, after analyzing the clusters in the K-means model, we notice a cluster clearly and largely detached from the others which could result in a very high MSE for one fold of the 5-fold cross-validation scheme in which the validation set would contain many of the points in that detached cluster.

For further investigation, one could analyze the performance of another regression model, such as K-medoids, or treat the targets (new hospitalized cases) as a linear function of features (symptoms) and perform linear regression on this model per region. Moreover, another interesting avenue could be to perform a separate regression model for each US-regions and compute the MSE for all of them. That way, one could assess if certain regions hold more of the responsibility for the noise in the dataset. The same strategy could be undertaken for smaller groups of symptoms, so as to identify those that are bigger cause of noise/inaccuracy.

5 Statement of Contributions

W.Bouchard and F.Beaupre contributed to every part of this project in pair.
A.Abouchedid did not contribute to the project, and thus elected to remove his name from the report.

References

[1] Search Trends Symptoms Dataset documentation, *Google Research*
https://github.com/google-research/open-covid-19-data/blob/master/data/exports/search_trends_symptoms_dataset/README.md

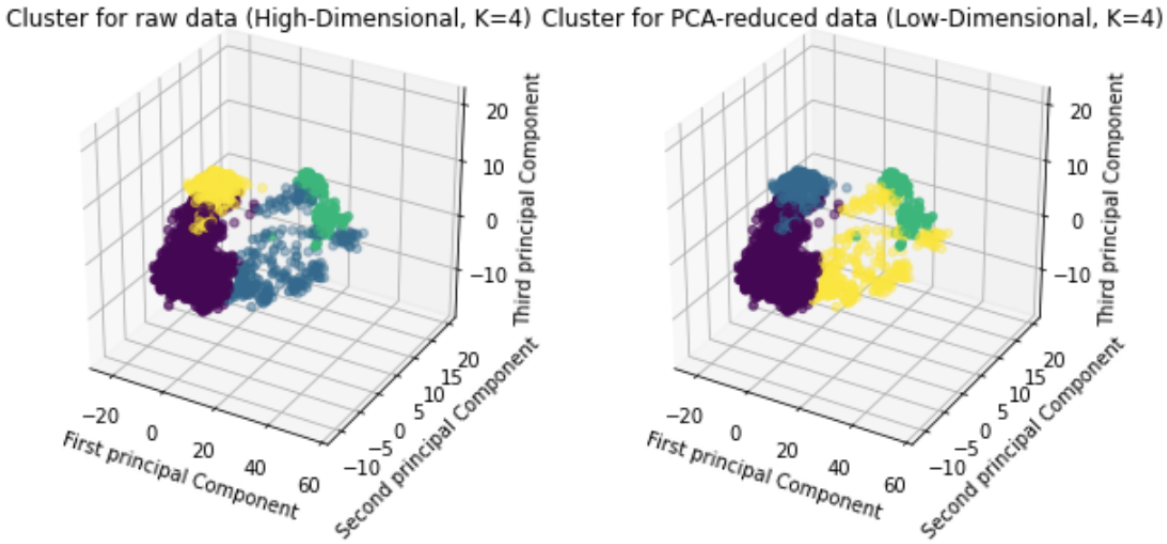
[2] COVID-19 Hospitalization Cases Dataset documentation, *Google Research*
https://github.com/google-research/open-covid-19-data/blob/master/README.md

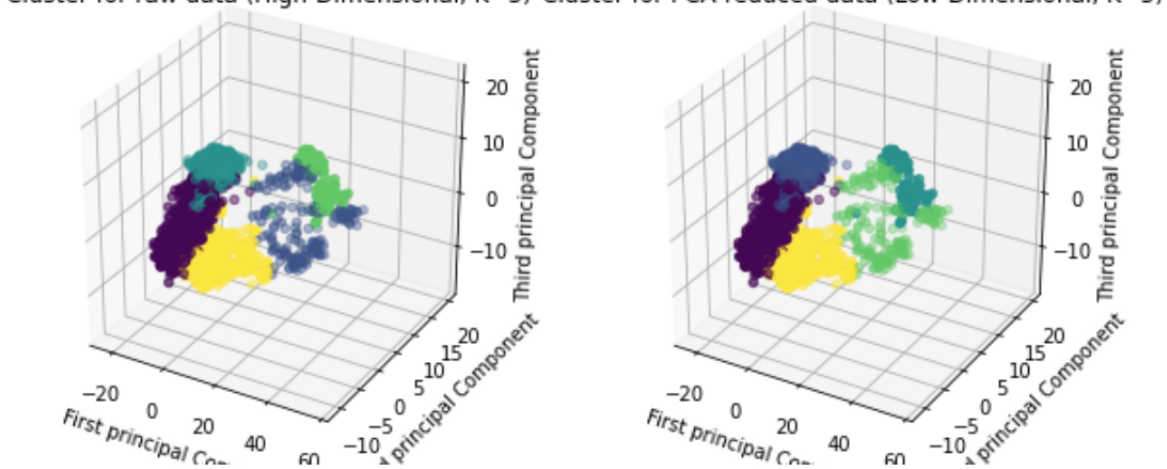
[3] Explore COVID-19 Symptoms Search Trends, *Google*
https://pair-code.github.io/covid19_symptom_dataset/?date=2020-02-17

[4] Marie-Claude Roy, M.D, Microbiologist-Infectiologist, CHU de Québec - Université Laval

6 Appendix A

Plots of the K-means model in 3D for $K = 4, 5$:





7 Appendix B

List of the covid-relevant symptoms that we used to reduce noise (description in the *Discussion and Conclusion* section) [4]:

- Chest Pain
- Chills
- Common Cold
- Confusion
- Croup
- Cough
- Diarrhea
- Fatigue
- Fever
- Headache
- Hyperthermia
- Low-grade Fever
- Migraine
- Muscle Weakness
- Myalgia
- Nasal Congestion
- Nausea
- Pneumonia
- Rhinorrhea
- Shivering
- Shortness of Breath
- Sore Throat
- Throat Irritation
- Syncope
- Upper Respiratory Tract Infection
- Vomiting
- Weakness

Note that according to expert Marie-Claude Roy [4], the symptoms Anosmia, Ageusia, Dysgeusia are the ones most heavily-related to COVID-19, but that the search trends dataset did not provide enough information for them, and thus we had to remove them from the dataset.