COMP 550 Mini Project 1 Report (Group 30)

James Berry (260629889), Nghi Huynh (260632588), and Krystal Xuejing Pan (260785873)

Abstract— **In task 1, we acquired, preprocessed, and analyzed both datasets. We found that roughly 20% of the features in the Search Trend dataset [1] and about 65% of the features in the COVID hospitalization cases dataset [2] are valid.**

In task 2, we found the search dataset's four most searched symptoms and visualized them using heatmaps. The search dataset was also reduced to a two-dimensional and a three-dimensional dataset using PCA. We clustered both raw and three-dimensional data by using K-Means clusters, and the best number of clusters (k=2) was found by using the elbow method. The clusters are visualized by a colored three-dimensional scatter plot.

In task 3, we implemented several regression models in order to predict regional hospitalization cases given the search trends data for the region. The models implemented were K-Nearest Neighbours (K-NN) and Decision Trees, as well as their related models Radius Neighbours and Random Forests.

I. Introduction

In task 1, we acquired, preprocessed, and analyzed both datasets. Before analyzing, we cleaned both datasets by removing any regions and features that exceeded a predefined threshold (any column with more than 65% of missing or invalid data entries). Then, we normalized both datasets. Finally, we converged the COVID hospitalization cases dataset from daily to weekly resolution before merging them into the search trends dataset.

In task 2, to most efficiently investigate the search dataset, the four most searched datasets were found by choosing the columns with the most overall normalized search values. They are aphonia, crackles, laryngitis and viral pneumonia. One heat map was created for each of them with respect to US open covid region codes and dates (Fig. 1.).

Then, the search data was reduced by using PCA. The best number of principal components is determined by visualizing the data's variance up to 20 principal components (Fig. 2.). Even though the PCA model with three principal components is chosen to proceed, both two-dimensional and three-dimensional models are plotted (Fig. 3.).

After, we clustered both raw (90-dimensional) and three-dimensional data with K-Means clustering. The best number of clusters is determined by the elbow method, namely trying values from 1 to 10 and plotting each model's inertia (Fig. 4.). Three is chosen to be the best value. We then plotted the 3d data with coloured clusters (Fig 5.).

In task 3, we implemented two different train-validation splits on the data, based on namely temporal and regional criteria – for the former data points obtained after 2020-08-10 were held as a validation set, whereas for the latter, a 5-fold cross-validation scheme was implemented, holding out 20% of regions as a validation set for each fold.

Models were then implemented for both K-NN and Decision Trees (using the Scikit-Learn library's implementations) and trained using hyperparameter values of 1-30 and 1-20 for the value of k and the maximum depth, respectively, for both train-validation split schemes. Finally, the Mean Squared Error of these models' performance on the validation data was plotted, and the best performing model hyperparameter value and its corresponding mean squared error were found.

Finally, the process above was repeated with variants of both frameworks. For K-NN, Radius Neighbours was used. This variant of K-NN searches for all data points within a particular radius r rather than searching for the k nearest data points to a new point to be predicted. For Decision Trees, we implemented the Random Forests model. This variant trains multiple decision trees on the training set, outputting the average prediction of them. The number of trees to be trained and their maximum depth as hyperparameters. We implemented models with varying hyperparameters as above before plotting their validation performance and obtaining the best hyperparameter values for both frameworks.

II. Datasets

For tasks 1.1 and 1.2, we downloaded both datasets by passing their raw hyperlinks, updated on Oct 9, 2020. Then, we loaded the datasets into Pandas DataFrames

and stored the Search Trends dataset and COVID hospitalization cases dataset as dataset_1 and dataset_2, respectively.

For task 1.3, we cleaned and normalized the datasets. First, we plotted the histogram of missing or invalid data entries for each dataset. We decided to set a threshold by observing the histograms where we dropped any feature exceeding 65% of missing data entries. Then, we normalized dataset_1 by region-specific for further analysis.

For task 1.4, we merged both datasets for further analysis. Since the time resolution of dataset_2 is daily instead of weekly, we first converted it into weekly resolution. Then, we normalized the weekly_dataset_2 and merged with normalized_dataset_1 using Pandas DataFrames.

For task 2.1, we first try to find how many unique region codes and dates there are. There are 16 unique region codes and 39 unique time points. Considering all the missing values that need to be dealt with, we found three methods: linear interpolation, mean and fill them all with zero [3]. We first denied the method of filling everything with 0 since it seems least faithfully represents the data. We found the paper *Comparison of Linear Interpolation Method and Mean Method to Replace the Missing Values in Environmental Data* [4]. The authors extensively compare the mean and the linear interpolation method when dealing with missing data and conclude that linear interpolation outperforms the mean method by a fair amount. Thus, we have employed the linear interpolation method to fill in the missing values.

The top four symptoms are then found by finding the four largest sums of all symptoms columns. For each of the top four symptoms, a new subset is created. The subsets only contain three columns: open covid region code, date and particular symptom search values.

For task 2.2, to apply PCA, we drop the columns with any data type other than float since we treat each time point and region separately. Two models of 2D and 3D are then created by using the reduced version of the search dataset.

For task 2.3, both the raw reduced version and the processed 3d version of the search dataset from the PCA model are clustered using K-Means.

For task 3, we took from the pre-processed merged dataset of task 1 the data points with data for our learning features (the symptom search data) - namely the points from U.S subregions. The features and labels (hospitalization rates) for our learning models were then extracted, and as in task 2, linear interpolation was used to fill in missing data. Finally, the feature and label data were both split using the two schemes – on a temporal criterion and a regional one, resulting in training and validation sets for the time-based split, as well as a 5-fold regional split to be used for cross-validation.

III. Datasets

Task 2:

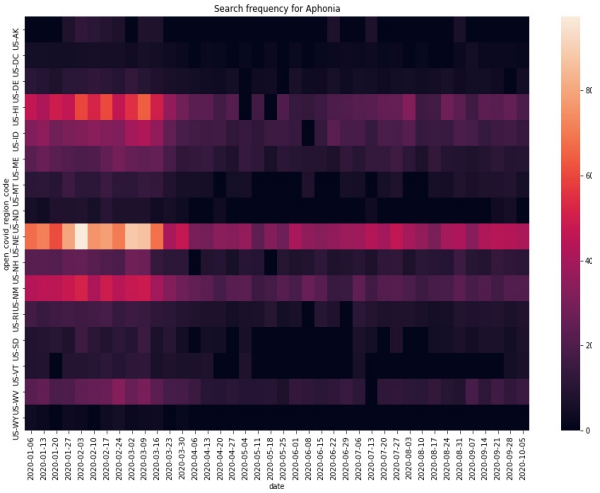

Fig. 1.A. Heatmap of **aphonia** with respect to open covid region code and dates

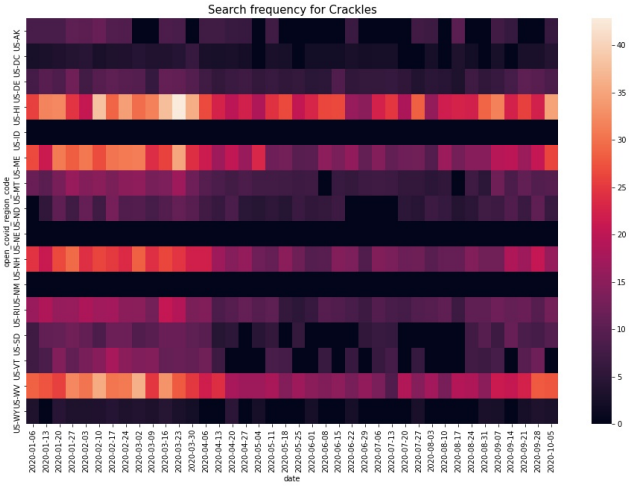

Fig. 1.B. Heatmap of **crackles** with respect to open covid region code and dates

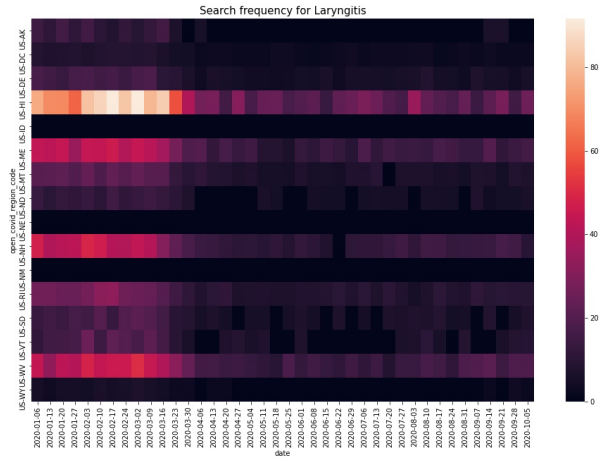

Fig. 1.C. Heatmap of **laryngitis** with respect to open covid region code and dates

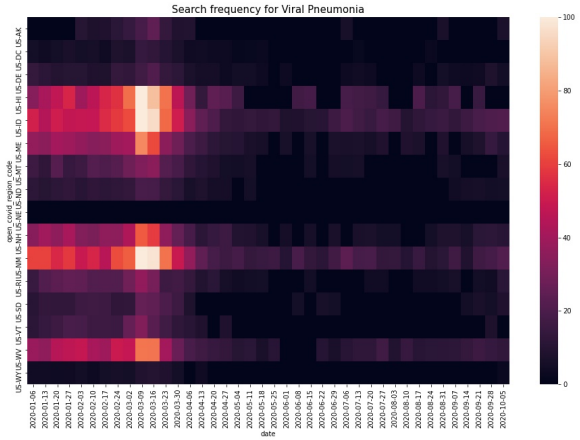

Fig. 1.D. Heatmap of **viral pneumonia** with respect to open covid region code and dates

Fig. 1. Is the collection of visualization of the search trend data for the top four symptoms: aphonia, crackles, laryngitis and viral pneumonia. It is clear in the heatmaps that for all four symptoms, there is a huge spike in search trend around March, particularly evident in certain regions, which presumably is relevant to local public health situations.

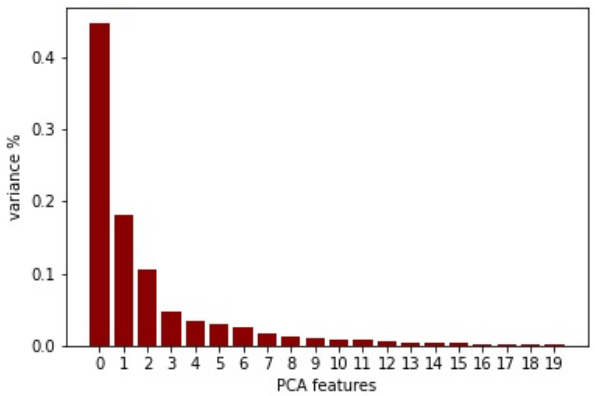

Fig. 2. This figure shows the changes in variances as the number of PCA features increase in this particular dataset. This graph helps us to determine that a 3D representation is necessary to more faithfully represent the dataset.

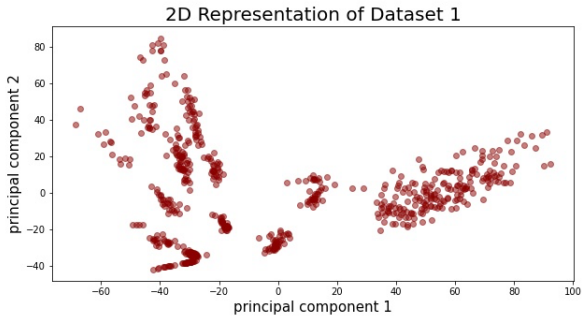

Fig. 3.A. Two-dimensional representation of dataset 1 after PCA

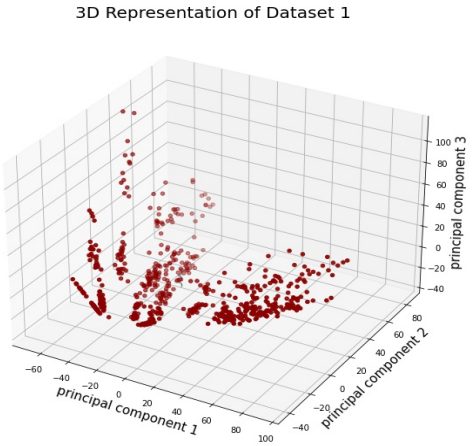

Fig. 3.B. Three-dimensional representation of the dataset after PCA

We choose the 3D version to proceed to clustering tasks.

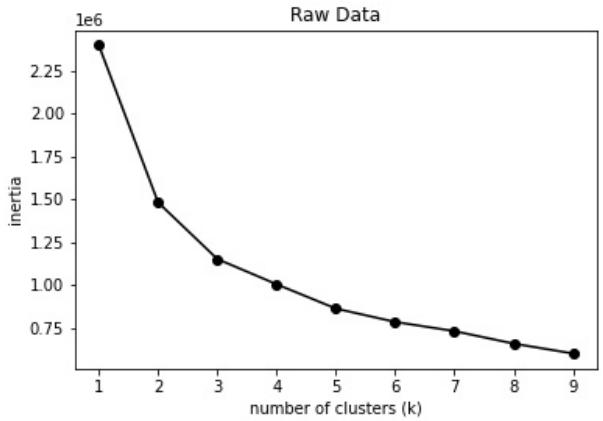

Fig. 4.A. the relationship between inertia and the number of clusters for raw data

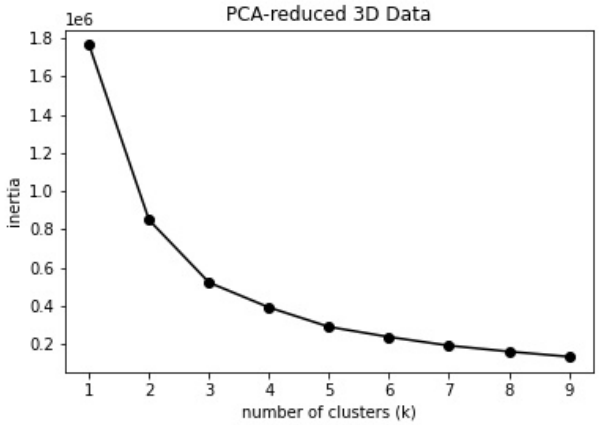

Fig. 4.A. the relationship between inertia and the number of clusters for PCA-reduced 3D data

To answer whether the clusters remain consistent for raw and PCA-reduced data (task 2.3), we have to find alternative ways since visualizing a 90-dimensional clustering is nearly impossible. However, we believe that it is possible to get a general idea from the visualized relationships between inertia and the number of clusters for both the raw and the 3d data (Fig. 4.). Inertia intuitively tells the distance between points in a cluster. The graphs show similar patterns, which implies that the clusters in both cases are similar.

However, this result is not convincing unless there is some theoretical basis. After some research online, we have found *K-means Clustering via Principal Component Analysis* [5] and *A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA* [6]. After extensive experimentations and theoretical analysis, both papers conclude that the results of clusters obtained using the PCA-reduced data were very close or even better in accuracy comparing to the high-dimensional original datasets. Thus, it is reasonable to conclude that the clusters remain consistent for the dataset's raw and 3d versions.

Task 3:

For K-NN the best performing model for the time-based split was found to be a K value of 2, with an average mean squared error of 0.01554 and an R2 score of 0.93982.

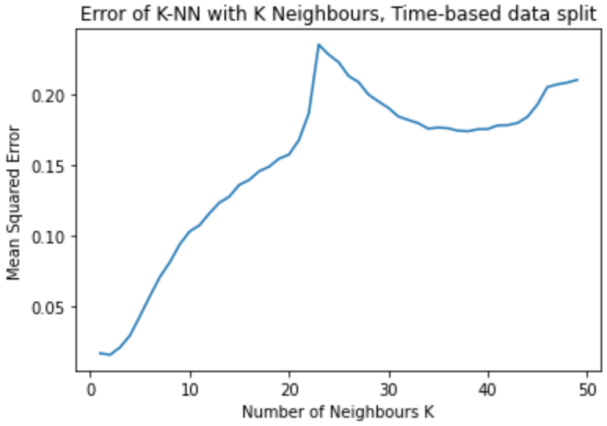

Fig. 5.A. the mean squared error performance of the models as K increase for K-NN model.

For Decision Trees, the best performing model for the time-based split was found to be a maximum depth of 4, with an average mean squared error of 0.01554 and an R2 score of 0.93982.

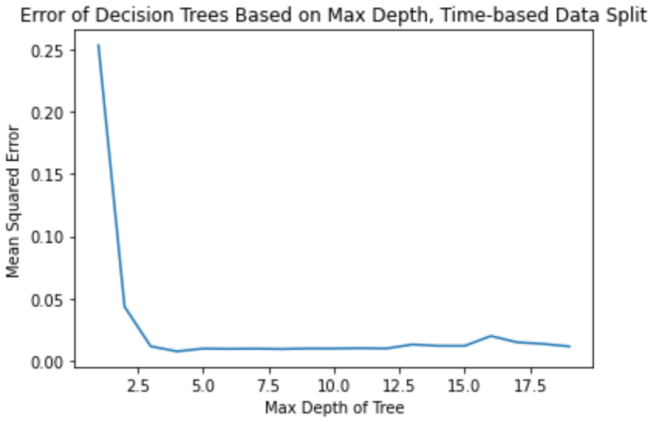

Fig. 5.B. the mean squared error performance of the models as K increase for decision tree model.

As can be seen, decision trees perform much better on the time-based split by both metrics.

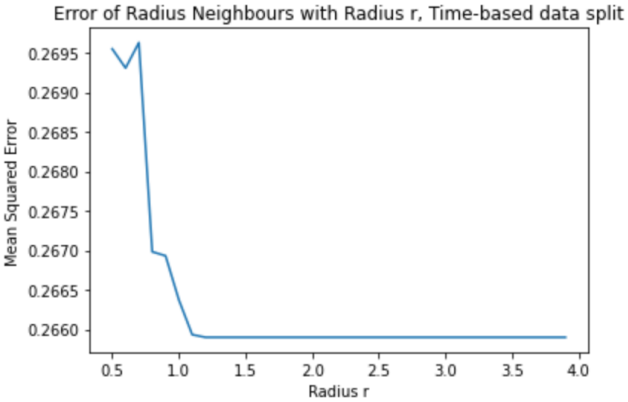

Fig. 5.C. the mean squared error performance of the models as K increase for radius neighbours model.

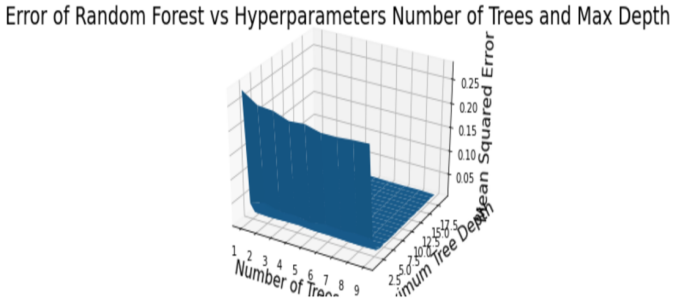

Fig 6. the error of rando forest vs hyperparameters numbers of trees and max depth

Finally, for the two variant frameworks, the best hyperparameter values were found to be as follows – for radius neighbours, a radius of 1.199, with a mean squared error of 0.26589 and an R2 score of -0.02950. And for random forests, two trees with a maximum depth of 5, with a mean squared error of 0.00751 and an R2 score of 0.97088. The performance of both frameworks as the hyperparameters vary is shown in Fig. 6.

Radius neighbours model performs extremely poorly, likely due to the minimal values of features in the dataset, thus small distances between data points. Random forests model performs much better than all three other frameworks, with distinctly better performance than K-NN and slightly better performance than a single decision tree.

V. Discussion and Conclusion

The takeaways from task 1 are data cleaning and normalization. It is essential to identify and remove incomplete parts of the data since coarse data causes unexpected results while modelling [7]. Moreover, data normalization will improve analysis from multiple models [8].

There are two critical takeaways from task 2. Firstly, principal component analysis is an effective way to visualize high-dimensional data in its entirety and prepare data for future analysis, such as clustering. However, the direct visualization of high-dimensional clustering should be explored further. Secondly, heatmaps are useful when visualizing the trend of data. Analysis of this sort can help people quickly understand the critical points in a set of data, and it could be applied in a real-time setting.

From task 2, we can see the importance of proper model selection during supervised learning tasks, with decision trees providing much better performance on the dataset than K-NN. Furthermore, we see the value of utilizing more developed models, with the more complex variant of decision trees and random forests, giving a distinct improvement in performance than a single decision tree. This choice of model can often be made by inference from the dataset, as here where the small values of features resulting in small distances between datapoints in feature space would indicate a likely poor performance with a radius neighbour model, but when not possible, we can see the benefit of training multiple model frameworks to compare performance before choosing a framework for the task.

Note on the region-based data split: Unfortunately, due to an unfound error in the regional split implementation, coherent results were not able to be obtained for the regionally split data, and so it is not present in this report.

Statement of Contributions

Nghi Huynh contributed task 1 and related writeup in the report.
Krystal Xuejing Pan contributed task 2, related write up in the report and report final formatting.
James Berry contributed task 3 and related writeup in the report.

References

[1] COVID-19 Search Trends symptoms dataset https://github.com/google-research/open-covid-19-data/blob/master/data/exports/search_trends_symptoms_dataset/README.md

[2] COVID-19 Hospitalization dataset https://raw.githubusercontent.com/google-research/open-covid-19-data/master/data/exports/cc_by/aggregated_cc_by.csv

[3] Working with missing data, pandas 0.12.0 documentation https://pandas.pydata.org/pandas-docs/version/0.12.0/missing_data.html

[4] Norazian Mohamed Noor, *et al.* "Comparison of Linear Interpolation Method and Mean Method to Replace the Missing Values in Environmental Data Set" August 2014.

[5] Chris Ding, Xiaofeng He, *et al.* "K-means Clustering via Principal Component Analysis"

[6] Maha Alkhayrat, Mohamad Aljnidi, Kadan Aljoumaa "A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA", Journal of Big Data 7, Article number : 9 (2020)

[7] Data Cleaning in Python: the Ultimate Guide (2020)
https://towardsdatascience.com/data-cleaning-in-python-the-ultimate-guide-2020-c63b88bf0a0d

[8] How to normalize dataframe pandas
https://www.kaggle.com/parasjindal96/how-to-normalize-dataframe-pandas