COMP 551 Project 1

Zhiying Tan, Ian Tsai, Jing Liu

October 22, 2020

Abstract

We investigated the performance of supervised and unsupervised learning on two COVID-19 datasets, a search trends dataset and a hospitalization cases dataset. We merged the two datasets in order to perform regression on hospitalizations using search trends. We visualized the search trend dataset by grouping the most popular COVID symptoms and graphed the symptom distributions by date and US state. To better understand the datasets, we visualized the different symptoms using dimensionality reduction and performed clustering to explore possible groupings of the search trends dataset. Both models gave a similarly low accuracy. We also incorporated weather data by state, which did not improve the accuracies. Future work could attempt to remedy this by using different regression models, or learning from different features.

1 Introduction

In this project we used K-Nearest Neighbors (KNN) and decision trees learning to tackle two COVID-19 datasets. Google Research's Open COVID-19 Data aggregates COVID-19 cases, deaths, hospitalization rates, etc., while [1] their COVID-19 Search Trends Symptoms Dataset aggregates search patterns for COVID-related symptoms at a weekly resolution within regions in the US [2]. We merged these two datasets in order to predict new hospitalizations using search trends. We present several visualizations of the search trend dataset, including symptoms distribution by region, a Principal Component Analysis reduced graph, and a K-Means clustering comparison. Upon performing regression with KNN and decision tree analysis, we found both to have a similar mean absolute error (MAE) on both a region-based split and time-based split.

Related works include [3], which examines features beyond the scope of this project such as seasonal behaviour, regional mortality rates, and effectiveness of government measures. Meanwhile, [4] includes different types of data such as medical images, sentiment analysis from social media. These, and other works, highlight the necessity of cross-domain research in effectively combatting COVID-19.

2 Datasets

2.1 Data Preprocessing

The datasets we used were last updated on 2020-10-20. First, we dropped columns that had all null values in both datasets, because they are essentially useless. We filtered the case dataset for US-only data, as the search trend dataset is geographically limited to the US. In the case dataset, we were left only with new and cumulative hospitalization rates, the labels we wanted to predict. To merge the two datasets, we converted the time resolution of the search trends data to weekly and changed the date for each point in the case data to match the closest date in the search data. The merged dataset has 121 features (symptoms) and 624 instances.

Additionally, since the search dataset uses a separate time factor for a given region and time resolution, we could not directly compare data across regions or time resolutions. As such, we standardized each symptom's popularity by region by removing the mean and scaling to unit variance. This is an important step, as KNN is affected by feature scaling. We also considered sklearn's `RobustScaler`, an oft-cited alternative to our `StandardScaler` that is sensitive to outliers and doesn't assume an underlying Gaussian distribution [5]. We also ran KNN and decision tree analysis with `RobustScaler` and found very similar results, detailed in Appendix A.

2.2 Distribution by region

We first visualized the search trend data by filtering out the top nine symptoms with the most complete data[1]. Though we chose the nine symptoms with the most complete data, we still ended up with many NaN values, which we ignored when plotting. Based on our distribution bar plots (Figures 1 and 2), aphonia and crackles seem to be the most popular and well-recorded symptoms.

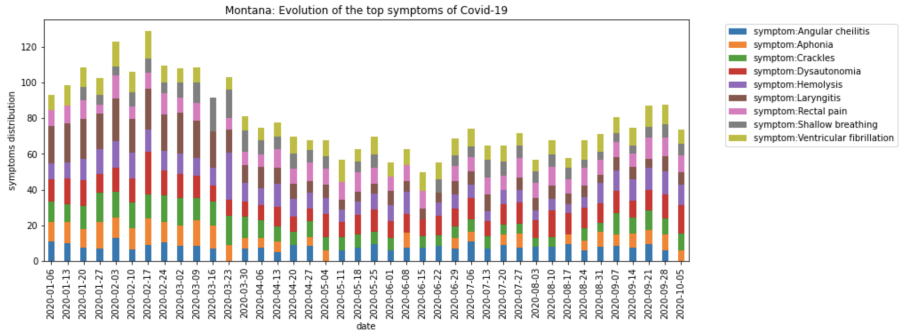

Figure 1: Example of top symptom searches in Montana, without missing data.

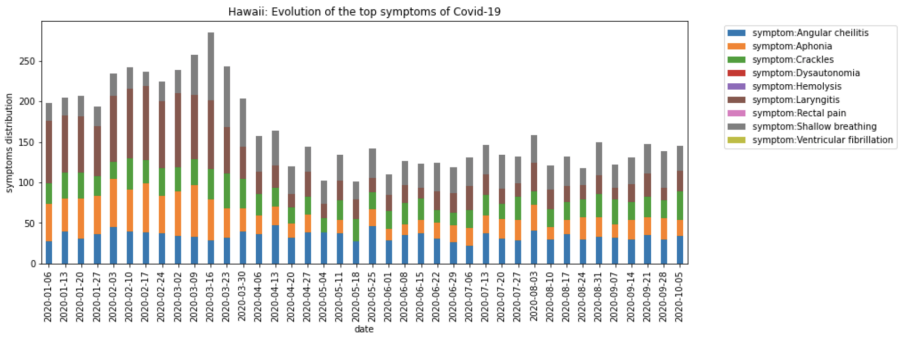

Figure 2: Example of top symptom searches in Hawaii, with missing data.

Plots for the remaining regions can be found in Section 2.1 in our code.

2.3 Feature Selection

We first cleaned features with more than 60% non-null features, as features with too many missing values will negatively influence the learning process. Then, we calculated the correlation between features and *hospitalized_new* to determine their significant levels. In this part, features with $|\text{corr}(feature_i, hospitalized\_new)|$

[1]However, some regions with missing data don't span the full distribution of the nine chosen symptoms; see our code for mode details.

≤ 0.01 are dropped. Since they are too low compared to the average correlation level, they may decrease the accuracy of our models.

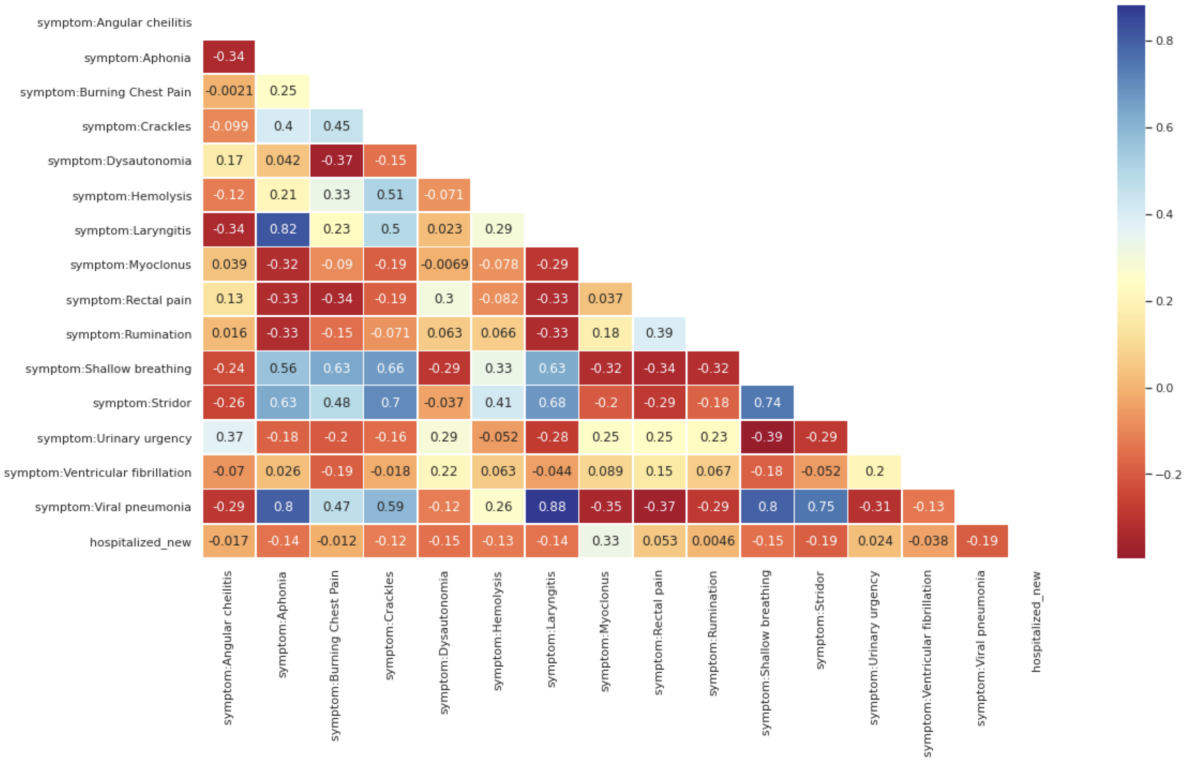

Figure 3: correlation

2.4 Feature Generation

Certain features are significantly positively and negatively correlated with *hospitalized_new*. We selected pairs of features from this group and took their product (e.g. $feature_i * feature_j$), regarding the products as new features if they are highly correlated to the prediction label. By doing so, we made the model more reliant on these more important features. In theory, this should lead to better performance, but trial runs of feature generation on KNN (detailed in Table 5 in the Appendix) did not support this hypothesis. Nonetheless, we implemented feature generation throughout both KNN and decision tree analysis and believe its potential can be realized with further experimentation (e.g. different combinations of features).

3 Results

3.1 PCA Visualization and K-Mean clustering

Figure 4 shows a PCA plot with the top five symptoms from the search trend dataset. First, we pivot the dataframe with dates recorded as index and symptoms as columns. We determined the number of components by plotting the percent variance explained as a function of number of principal components, and found the elbow point at 2 components (Figure 6 in the Appendix). We use sklearn's `PCA` to reduce the dimensionality of the original data.

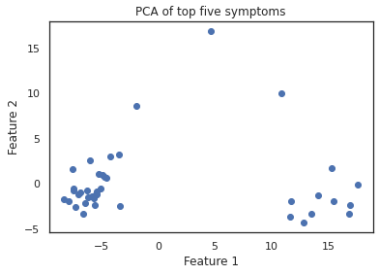

Figure 4: PCA plot of the top five symptoms.

Figure 5 shows the K-mean clustering comparison between a PCA reduced data and its original data. When working with the reduced data with k=5, since we are working with the top five symptoms, the clustering looks somewhat promising–most clusters tend to follow their centroids.

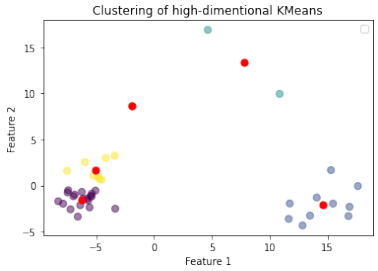

Figure 5: Reduced PCA plot of the top five symptoms data, using K-Means clustering. The red dots indicate the cluster centroids.

3.2 Comparison of regression performance

We applied KNN regression using sklearn's `GroupShuffleSplit`, which separates cross-validation folds by labelled groups (in this case, US state). We also performed a parameter search for $k = [0, 100]$ as a function of MAE, in order to find the best-performing hyperparameter k. We performed 5-fold cross validation on the data, though it should be noted that we did not separate a test set beforehand[2]. The downside of this method is that there will be an increase in bias, as the test data is effectively seen prior to testing. As such, our results may be artificially high.

Before building the decision tree regression model, we applied sklearn's `RandomizedSearchCV` algorithm to figure out the optimal combination of hyperparameters. We choose one of the optimal models with smallest pair of max_depth and max_leaf_node to avoid overfitting problem. Then we split the data by 5-fold cross validation algorithm and train the model to get 5 groups of predicting result.

Average MAE		
Result	KNN	Decision Tree
MAE, region-based split	25.755	30.939
MAE, time-based split	32.047	31.948

Table 1: Average MAE for KNN and decision tree learning. $k = 5$ and 3, respectively, for the two KNN dataset splits.

Table 1 shows the MAE of KNN and decision tree learning. KNN is deterministic within the given constraints: in the region-based split, the same groups are used each time for the cross-validation, and in

[2]a method approved by the TAs in office hours.

the time-based split, all data points before August 10, 2020 are partitioned to training, while all points after are used in test. Decision trees are similarly deterministic, so one run suffices for each method of regression.

KNN performs better than MAE, more so for the region-based split than the time-based split. However, the MAEs are high across the board, indicating that neither regression method may have been well-suited for the task, or, alternatively, that the dataset is not well-suited for regression. Further investigation could include different ways of normalizing the data, or different ways of dropping means.

3.3 Further Experiments

We wanted to see if higher temperatures would reflect a lower total rate of COVID-19 cases. To examine this hypothesis, we scraped a US climate dataset by state, where the cited temperatures are the monthly highs of the state capital [6].

After preprocessing the new dataset and integrating it with the original features, we found that the correlation between temperature and the predicted label is small. Tables 2 and 3 compares the MAE between the two models, for decision trees and KNN. As the additional feature of temperature doesn't significantly impact our errors, we decided not to include this factor in our final model. However, further work using temperature could incorporate both monthly highs and lows, or various regions inside a state, which may have diverse climates.

MAE (Decision Tree)		
Result	original	original+temperature
MAE, region-based split	30.939 $\pm$ 3.55	29.428
MAE, time-based split	31.948	33.311

Table 2: Impact of including temperature on average MAE of decision tree, using 5-fold cross validation

MAE (KNN)		
Result	original	original+temperature
MAE, region-based split	25.755	64.027
MAE, time-based split	32.047	26.927

Table 3: Impact of including temperature on average MAE of KNN, with 5-fold cross validation.

4 Discussion and Conclusion

In conclusion, after visualizing the search data in two dimensions, we found that KNN and decision trees were unable to accurately predict new hospitalization cases using symptom search trends. We explored the implications of feature engineering, selection, and generation, as well as different methods of normalizing data across regions–these returned insignificant results, but could be explored in further depth. Future work could better methods of feature selection (e.g. Pearson residuals), different methods of feature engineering (transforming some numeric values to categorical to reduce noise), better hyperparameter optimization, different NaN handling, and different clustering methods.

In addition, we also performed regression using a weather dataset, but did not find significant differences in error. Further exploration using a more accurate weather dataset would be worth pursuing, as well

as other datasets (e.g. those mentioned in the Introduction). Finally, more sophisticated methods of regression such as neural networks should return smaller errors.

5 Statement of Contribution

All team members contributed to task 1, Ian primarily wrote task 2, and Jing and Winnie wrote task 3. All members contributed equally to the report.

References

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

[2] ——, "Google covid-19 search trends symptoms dataset." [Online]. Available: http://goo.gle/covid19symptomdataset 1

[3] Open data resources for fighting covid-19. Accessed Oct. 20, 2020. [Online]. Available: https://www.researchgate.net/publication/340644288_Open_Data_Resources_for_Fighting_COVID-19 1

[4] Covid-19 open source data sets: A comprehensive survey. [Online]. Available: https://www.researchgate.net/publication/341654437_COVID-19_Open_Source_Data_Sets_A_Comprehensive_Survey,note= 1

[5] "Feature scaling with scikit-learn." [Online]. Available: https://benalexkeen.com/feature-scaling-with-scikit-learn/ 2

[6] Climate data. Accessed Oct. 20, 2020. [Online]. Available: https://www.usclimatedata.com 5

A Supplementary Data and Plots

Result	KNN	Decision Tree
MAE, region-based split	25.003	33.403
MAE, time-based split	31.003	33.311

Table 4: Average MAE for KNN and decision tree. $k = 5$ and 3, respectively, for the two dataset splits. Data standardized with sklearn's `RobustScaler`.

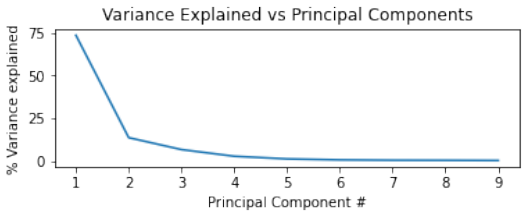

Figure 6: The figure illustrates the elbow point of the number of principal components.

Result	KNN, original	KNN with Feature Engineering
MAE, region-based split	25.987	25.755
MAE, time-based split	31.388	32.047

Table 5: Impact of feature engineering on average MAE for KNN. For original KNN, $k = 7$ and 2, respectively, for the two dataset splits. For KNN with feature generation, $k = 5$ and 3, respectively. Data standardized with `StandardScaler`.