COMP 551 - Mini Project One

Group 6 - Dean Meluban, Javier Ordenes Alegria, Ryan Wilson

1.0 Abstract

In this mini project, our group investigated the performance of two regression models, including K-nearest neighbors (KNN) and decision trees. The main objectives were to visualize raw and dimension-reduced data trends and predict hospitalization cases from search trends for COVID-19 symptoms within various sub-regions of the United States. Because of the number of variables, a principal component analysis (PCA) was performed to reduce dimensionality of the data, and better classify the data structure. Clustering was optimized using the elbow method to establish an appropriate number of clusters and minimize the computational cost of the algorithm. Performance of the K-means algorithm was lower than expected, possibly due to low correlation factors observed between the variables. It was found that KNN regression modelling performed better overall than the decision tree technique based on validation and root mean squared error (RMSE) metrics. Decision tree modelling achieved slightly improved training scores, but this is likely related to overfitting, as is common to the method. Neither technique demonstrated significantly faster training times as both require a certain level of hyperparameter tuning and/or pruning. Both KNN and decision tree models indicated that a region-based cross-validation (CV) approach is more effective than a time-based split for the explored datasets. Obvious drawbacks of the final merged dataset, including a lack of search trends for well-known COVID-19 symptoms (e.g., dry cough, fever), limited the analysis and other potential inferences.

2.0 Introduction

Key objectives for the project include the exploration of visualization and regression techniques using data related to internet search trends of COVID-19 symptoms and potential links to new hospitalization cases. The final database was derived from two separate open datasets ('Covid-19 Weekly Search Trends Dataset' and 'Open Covid-19 Dataset') obtained through Google Research and subsequently filtered, merged and cleaned for further processing and analysis.

Data visualization was carried out by comparing raw and dimension-reduced data trends using K-means clustering. Dimension reduction was achieved by Principal Components Analysis (PCA), which is a multivariate technique that attempts to extract the most important information by forming orthogonal linear combinations of the original variables and identify similarities among observations [1]. The elbow method was used to optimize the number of K-means clusters and minimize computational cost. Clustering was less effective than expected, possibly due to low correlation among the different variables.

Regression modelling was conducted using two separate supervised learning techniques, including K-nearest neighbor (KNN) and decision tree modelling, in an attempt to predict hospitalization cases related to internet search trends for COVID-19 symptoms. The KNN algorithm is considered a non-parametric 'lazy learner' because it simply memorizes inputs and predicts labels by finding the most similar examples in the training set [2]. Decision trees represent another non-parametric method used for both classification (categorical) and regression (continuous). Decision tree models are so named because they are built in a tree-like structure, breaking data down into smaller subsets (from internal decision nodes to terminal leaf nodes) based on some criterion (e.g., entropy, Gini index, MSE) [3]. The KNN regression model outperformed the decision tree algorithm, with higher validation scores and lower RMSE values overall. Higher training scores were observed in the decision tree model, but this is likely related to overfitting which is common to the method. Both KNN and decision tree techniques suggest that a region-based CV split is more appropriate than a time-based approach for the dataset.

3.0 Datasets

The datasets used were the Google Research Covid-19 Weekly Search Trends Dataset (hereafter referred to as the symptom set) and the Google Research Open Covid-19 Dataset (under the CC-BY license, hereafter referred to as the hospital set).

Various filtering techniques were used on the symptom set in order for learning models to be effectively utilized. First, all features that did not have any data were deleted. Rows in which over half of the features were null were also dropped using *pandas* library modules (dropna()). The index was set as a multi-index, using both [‘open_covid_region_code’] and [‘date’]. For the hospital set, it was noted that the only important feature was the ‘hospitalized_new’ feature. Therefore, hospitalized_new features were grouped by week, in order to properly align with the symptom set. An equivalent multi-index [‘open_covid_region_code’, ‘date’] was set for the subsequent merge.

The datasets were merged using both an inner join and a left join (left as symptom set) in order to align the dates and their corresponding weekly hospitalization totals. The symptom set was then normalized using the *scikit-learn (“sklearn”)* MinMaxScaler in accordance with regions [Alaska, Montana, North Dakota, South Dakota, Vermont, Wyoming]. A final preprocessing step included the removal of any features containing less than 50% (+/- 0.5%) data points in order to ensure data analysis and predictive modelling were carried out on sufficiently dense actual data (rather than imputed). Using the inner join, the final database includes a total of 130 sample sets in 6 states (sub-regions) with 100 features (symptom search trends) and 1 target variable (new hospitalization cases).

4.0 Results

4.1 Data Visualization and Clustering

Prior to data visualization, all missing values were imputed using the mean value for each variable (‘df.fillna(df.mean()). This step was not completed as part of the initial data preparation and preprocessing as it was decided that imputation for the regression modelling (Task 3) should be completed following the generation of cross-validation splits (see Section 4.2).

The second step was to generate different visualizations of the raw data for a subset of key search symptoms. These symptoms were selected using a 40% threshold (relative to the target variable) established from a correlation matrix, and included ‘Adrenal Crisis, ‘Ageusia’ and ‘Dysgeusia’. A dynamic visualization was also coded, in which the evolution of symptom search trends can be observed over time. Additionally, a scatter plot of raw data vs target variable (Figure 1) was performed to visualize the cluster distribution prior to dimension reduction and K-means clustering.

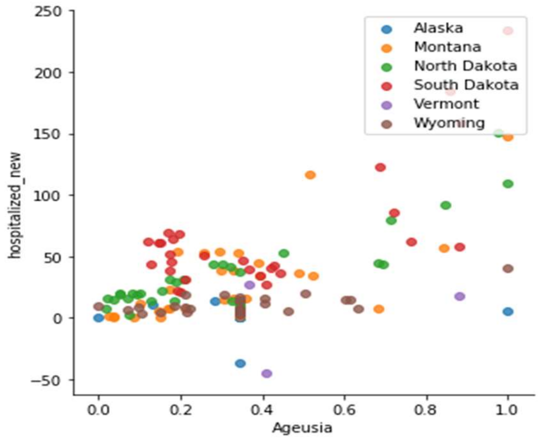

Figure 1: Scatter plot of "Ageusia" symptom search vs target variable "Hospitalized New".

The dimension reduction was also used to visualize the relationship between features and the target variable. To this end, the scores of the principal components were plotted to visualize the interaction between the reduced dimension data structure and the target variable "hospitalized_new" (Figure 2).

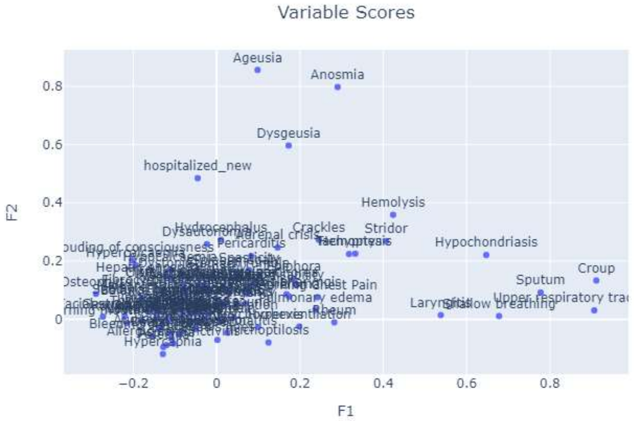

Figure 2: Scatter plot of variable scores

Dimension reduction through Principal Components Analysis (PCA) was carried out in order to train the K-means algorithm. We need to determine the appropriate number of principal components to extract in order to achieve the most interpretable data structure. Eigenvalues are first calculated for each factor to establish which provide the most information (i.e., explain variance).

A scree plot was coded in order to select the number of components to keep for subsequent training. The eigenvalues are a measure of the amount of variance accounted for by each factor and are therefore useful in determining the appropriate number of factors to extract. Eigenvalues are plotted, and a line is drawn through each (starting from the highest value) in order to observe where the slope of that line falls sharply. When the difference between two consecutive factors is very low, an additional factor would add relatively little to the information to that already extracted by the previous factors. From that point on, the amount of additional variance explained is insignificant, and therefore successive factors are excluded from the analysis. Below is the graph with which it was determined that the PCs to be used are PC1 and PC2, which accumulate most of the accumulated variance (Figure 3).

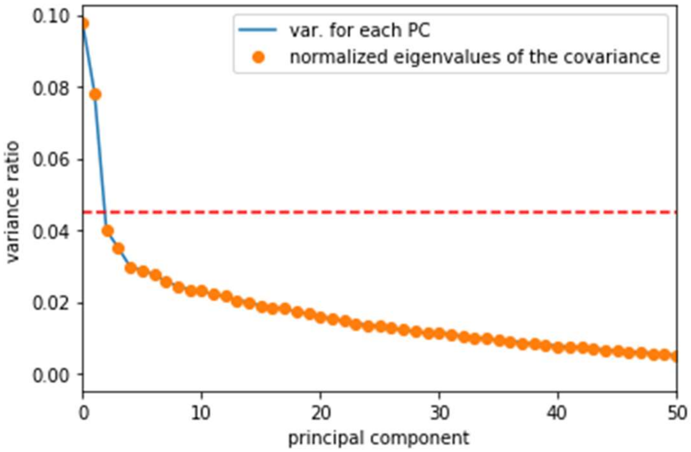

Figure 3: Scree Plot Eigen values principal components.

Based on the main components selected in the previous subtask, a new data structure is generated to classify the symptom search data to train the K-means algorithm. The K-means clustering method is an unsupervised machine learning technique used to identify clusters of data objects in a dataset. A cluster refers to a collection of data points aggregated together because of certain similarities. K-means algorithm identifies 'K' number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The main element of the algorithm works by a two-step process called expectation-maximization. The expectation step assigns each data point to its nearest centroid. Then, the maximization step computes the mean of all the points for each cluster and sets the new centroid. After choosing a number of clusters and the initial centroids, the expectation-maximization step is repeated until the centroid positions reach convergence and are unchanged.

For the selection of the number of K, an evaluation was carried out using the elbow method (Figure 4). This mathematical optimization is used to find a cut-off point to determine the correct number of clusters, and represents a common method to identify the point at which diminishing returns are no longer the additional cost. In clustering, this means one should choose a number of clusters so that adding another cluster doesn't give much better modelling of the data.

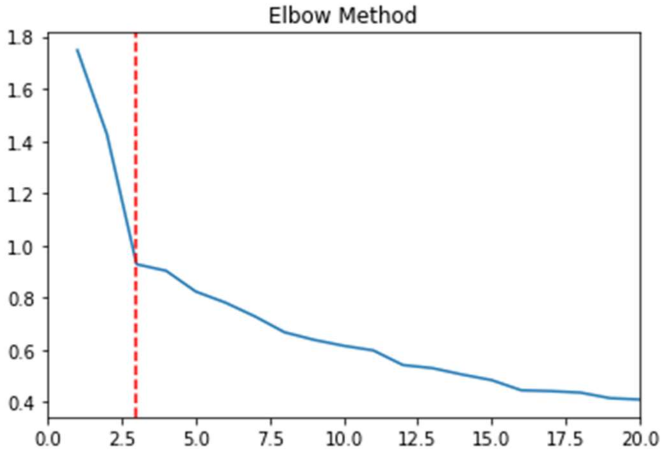

Figure 4: Elbow method result, K=3.

From this optimization it was determined that the optimal number is K=3. The result of the visualization from the main components manages to classify the data into 3 defined clusters. For the google search data, the results of the clustering based on the code performed in Task 2 are shown in Figure 5.

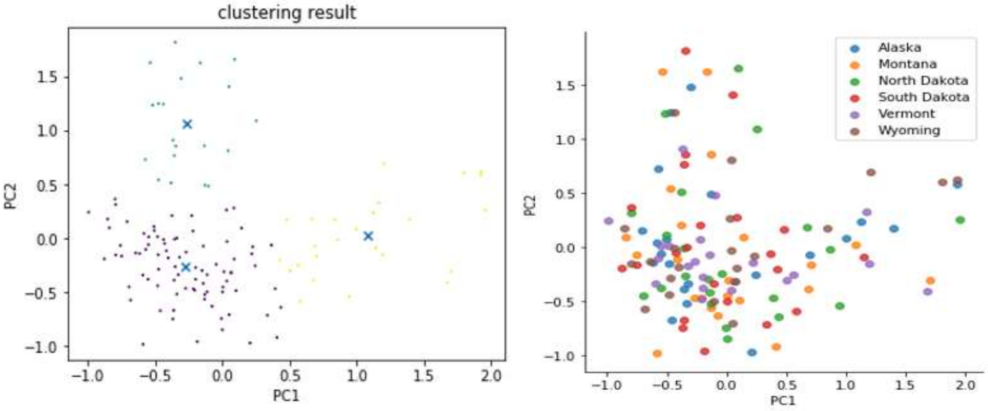

Figure 5: Clustering based on Principal Components (PC1 vs PC2).

4.2 Supervised Learning

Two supervised learning techniques, including K-nearest neighbors (KNN) and decision tree modelling, were employed to predict hospitalization cases given the search trends data. For this portion of the study, the data was split into training and validation sets using both region- and time-based strategies in order to estimate cross-validated model performance.

Provided the final database established from preprocessing/merging only contained data spread across 6 states, it was decided to split the data into corresponding subsets such that each region could be held out as a validation set as part of a 6-fold cross-validation (CV) region-based strategy. For the time-based split, a cut-off date of '2020-08-10' was used to establish training (pre cut-off) and validation (post cut-off) sets.

Following the splits, missing data values were imputed using the mean of each training and validation set. This ordered approach was used in order to avoid the 'bleeding' of information from the validation sets into the training sets that would have occurred had imputation been carried out prior to data splitting [4].

K-Nearest Neighbor (KNN) Modelling

The present study incorporated the use of the *sklearn* KNeighborsRegressor algorithm to fit a KNN model to each training set and predict hospitalizations in the corresponding validation set. This regression algorithm operates similarly to the typical classification algorithm, except that it can handle continuous input feature distributions (i.e., symptom search trends) and uses the mean of the K-neighbors to predict the target values.

For each training and validation set, the algorithm was run through a loop of potential K-neighbors to include, ranging from 1 to 25. Various metrics were computed, including training and validation 'scores' (i.e., coefficients of determination, R^2), as well as the root mean squared error (RMSE) between predicted and observed validation response variables (Tables 1-4). These metrics are useful to assess overall model performance, and help with fine-tuning of the K-neighbors hyperparameter.

For the region-based model, training performance was similar across all cross-validation sets, with an overall mean R^2 value of 66% (Table 1) but decreased by up to 50% with increasing K-neighbor value. Overall validation performance was very poor, with an overall mean R^2 value of -134%. Though disappointing, this is not necessarily surprising as KNN models predict based on similar training examples and we are trying to predict count-based hospitalizations for a volatile disease. Though discrete by definition, the target variable can effectively be considered 'continuous' and therefore is rather difficult to estimate/predict exact class numbers based on the similarity of internet search trends alone. As a result, it is much more effective to evaluate RMSE values in order to monitor model performance and draw inferences. Despite this, the highest validation scores (approx. -85 to -80%; Table 2) were clearly observed for K-neighbor values in the range of 5 to 8. These K-neighbor values also correspond to the lowest RMSE values (25.8-26.5; Table 3), and as such K=7 (RMSE = 25.8) is selected as the best K-hyperparameter for the final model. Figure 6 shows representative plots of the final model performance from CV set #3, including line plots of the training/validation scores and RMSE (all K-values) and a scatter plot of validation vs. predicted hospitalizations (K=7).

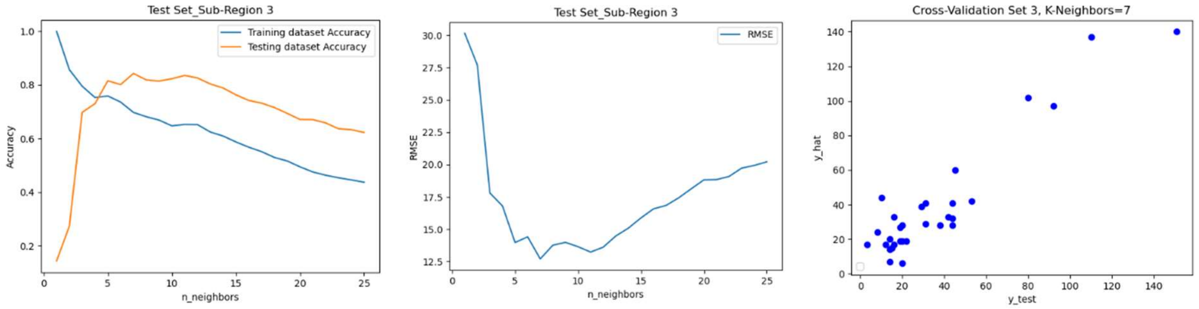

Figure 6. Representative plots of final region-based KNN model performance from CV set #3: a) line plot showing tradeoff between training and validation scores with increasing K-value; b) line plot of RMSE values shows global minimum at K=7, and; c) bivariate plot of observed validation vs. predicted hospitalizations (K=7).

K-fold CV was not possible for the time-based modelling strategy as only 1 training and validation set was generated using a static cut-off date of 2020-08-10. The same KNN regression algorithm was run through a similar loop of potential K-neighbor values ranging from 1 to 25. Compared to the region-based strategy, the time-based model underperformed across each of the computed metrics, with overall mean values of 46% and -262% for training and validation scores, respectively, and 37.64 for RMSE (Table 4). However, a similar inverse relationship between training scores and the number of K-neighbors is observed. This behaviour is expected as KNN models tend to overfit the data at low K-values (e.g., for K=1, the search space is restricted to the actual data point itself) [5]. In terms of fine-tuning the K-hyperparameter for the time-based model, the lowest RMSE values (~33.8-34.3; Table 4) are clearly observed in the K=3 to K=5 range, which also correspond to the highest training scores (-199 to -191%). Due to decreasing training scores, the final model should likely select a K-value of 3 or 4 in this case. Representative plots are not provided herein for the time-based strategy but are available in the provided source code.

Overall, the data supports the hypothesis that a region-based model should outperform a time-based strategy based on the volatile nature of COVID-19 outbreaks. Because transmission of the disease is mainly related to increased human contact and proximity, regional trends should provide greater relevance and insight towards the prediction of potential hospitalizations. However, a time-based strategy may provide useful information as a sub-model *within* each region by testing varied cut-off dates around known periods of disease outbreak.

Decision Tree (DTree) Modelling

The current work used the *sklearn* DecisionTreeRegressor algorithm to fit a decision tree model to each training set and predict hospitalizations in the corresponding validation set. Similar to the KNN portion of the study, the algorithm was run through a loop of potential maxDepth values (ranging from 1 to 25), and the same series of metrics were computed, including training and validation scores (R^2) and RMSE values (Tables 5-8). Again, these metrics were used to evaluate overall model performance and 'prune' the DTree model through hyperparameter adjustments. Additional adjustments were made to reduce instability and overfitting by increasing the minimum number of samples for each internal (decision) node split and the minimum number of samples to form a new leaf node. Through trial and error, the final model was fit using values of 10 and 5 for these hyperparameters, respectively.

For the region-based strategy, training performance was again similar across all cross-validation sets, with an overall mean R^2 value of 80% (Table 5) but increased by nearly 40% with increasing maximum depth. This trend was expected as decision trees are prone to overfitting when tree depth is not constrained. Validation performance was even worse than for the KNN model, with an average mean R^2 value of -3.51 (-351%). The highest validation scores (~26.4-26.9%; Table 6) are associated with a maxDepth range of 3 to 4, which also corresponds to the lowest RMSE values (27.92-27.97; Table 7). Thus, a maximum tree depth of 3 was selected to fit the final model. Figure 7 shows representative plots of the final model performance from CV set #3, with line plots of training and validation scores across all tested depth values and a scatter plot of validation vs. predicted hospitalizations for a maxDepth value of 4.

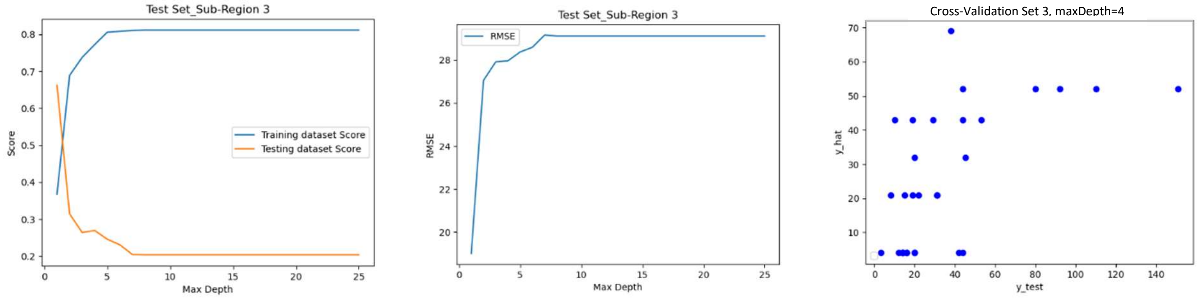

Figure 7. Representative plots of final region-based DTree model performance from CV set #3: a) line plot showing tradeoff between training and validation scores with increasing K-value; b) line plot of RMSE values shows local minimum at K=~3-4, and; c) bivariate plot of observed validation vs. predicted hospitalizations (K=4).

Compared to the region-based model, the time-based strategy performed marginally better in terms of training scores (mean R^2 value of 87.2%), similarly with respect to validation scores (-337%) but underperformed significantly for the RMSE metric (41.46) (Table 8). A maximum depth value of 2 was selected for the time-based model based on the highest validation score (-330%) and lowest RMSE value (41.14) among tested candidates (Table 8). Representative plots for the time-based strategy are not provided herein but are available in the attached source code.

Overall, the KNN models (Tables 1-4) performed better than the decision trees (Tables 5-8) based on validation and RMSE metrics for both region-based and time-based cross-validation techniques. Though training scores were higher for the decision tree modelling, this likely indicates overfitting based on poorer overall model performance. Both regression model types generated the best combined validation and RMSE results for CV sets 3 and 4, which hold out Montana and North Dakota as validation sets, respectively. This could be related to more sparsely distributed clusters associated with these sub-regions (as identified in Task 2), which would diminish overall predictive power when included in training sets.

5.0 Discussion and Conclusions

Following analysis of the preprocessed data sets, there is substantial evidence that the models do not perform as well as one would expect, particularly in comparison to examples seen in class (i.e., iris flower dataset). Hypothesis for why the data yielded such substantial errors include the omission of relevant COVID symptoms, such as dry cough and fever. Furthermore, due to the nature of the grouping in preprocessing, COVID 'hot-spot' regions where there were large outbreaks in the United States (such as New York, Florida, California, and Texas) were omitted, while low population areas of the United States (North Dakota, South Dakota, Wyoming, etc.) remained in the dataset. This perhaps provided a poor sample space for regression techniques, yielding higher than normal RMSE for respective techniques (KNN and DTree).

Analysis yielded better results when models were implemented using region-based splitting, which could be explained through the nature of the disease – there is correlation between regions getting infected due to virus being spread through close contact. It could be interesting in the future to explore time-based sub models *within* distinct regions of the dataset, clustered around time intervals in which outbreaks are defined and well-known. Our group has decided that this would lead to better regression models and more accurate predictions in future studies.

6.0 Statement of Contributions

Dean Meluban
Task 1: Data acquisition, preprocessing, filtering/merging, feature normalization, report writing.

Javier Ordenes Alegria
Task 2: Data visualization, K-means clustering, PCA modelling, testing/analysis, report writing.

Ryan Wilson
Task 3: Selection of train/test sets, imputation, predictive modelling, testing/analysis, report writing.