COVID-19 Awareness & Cases in Ohio State

Wonha Shin and Japnit Singh

***Abstract*— This report delves into the relationship between various socioeconomic, educational, and awareness variables and the spread of COVID-19 in Ohio's counties. A comprehensive analysis of 144 variables ranging from health metrics to topic awareness—measured through methods like Jaccard similarity—aims to present a multi-faceted view of the pandemic's footprint. A Random Forest model was employed, achieving an R^2 score of 0.95 and approximately 89% accuracy, to predict COVID-19 cases using key indicators such as population, education level, and health insurance coverage. The analysis revealed that topics like sports and entertainment remained focal points of public interest throughout the pandemic, possibly overshadowing critical health information. Geographical and temporal assessments provided insights into how awareness fluctuated over time and varied across counties, with significant disparities noted in disease spread and mortality rates. The study underscores the need for targeted communication strategies to enhance public health awareness and response in the face of such crises.**

I. INTRODUCTION

This lab report explains about the various columns like labor_force and awareness variables and how (or if) they have a significant impact on the number of COVID-19 cases in Ohio. Through analysis we aim to uncover patterns of correlation between number of cases and other attributes in the dataset. Understanding the data and its attributes is vital for selection of columns that need to be trained.

We trained the data against Random Forest to get around 0.89 accuracy on the data set with an r2_score of 0.95.

II. DATA

A. Key Variables

- COVID-19 Data: cases and deaths columns provide numerical data on the COVID-19 impact per county.
- Topic Awareness: Variables like core_ jaccard, core_ cosine, social_ jaccard, and politics_ jaccard suggest a focus on measuring topic awareness, possibly related to how well-informed the population is regarding various subjects.
- Socioeconomic Indicators: A variety of columns provide socioeconomic data, such as labor_ force _ rate, unemployment_ rate, median_ housing_ cost, median_ household_ earnings, and poverty_ rate, among others.

B. Descriptive Analysis

The dataset's broad range of variables offers a multi-dimensional view of the COVID-19 pandemic's impact across Ohio's counties. With data on 144 variables spanning health, societal, and economic sectors, we tried to analyze it to provides a holistic understanding of the pandemic's effects. In addition to tracking the virus's direct impact through cases and deaths, we delved into how communities interacted with various information topics, revealing differences in engagement and awareness.

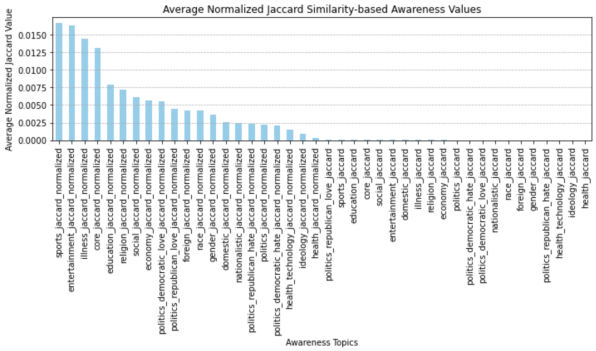

Fig. 1. Average Awareness Values.

The analysis(Fig. 1) highlighted that sports and entertainment were the topics with the highest average normalized Jaccard similarity-based awareness values, suggesting that these areas maintained significant public interest or media focus during the pandemic period. Conversely, topics related to health, ideology, and health technology garnered less attention, as indicated by their lower average values.

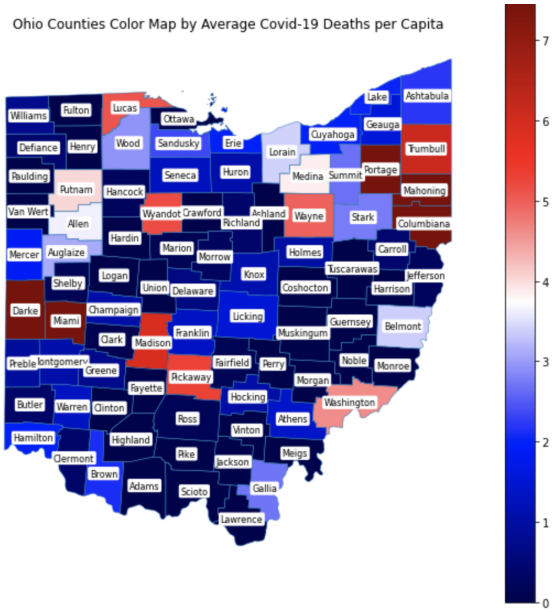

Fig. 2. Color Map Average Cases per Capita

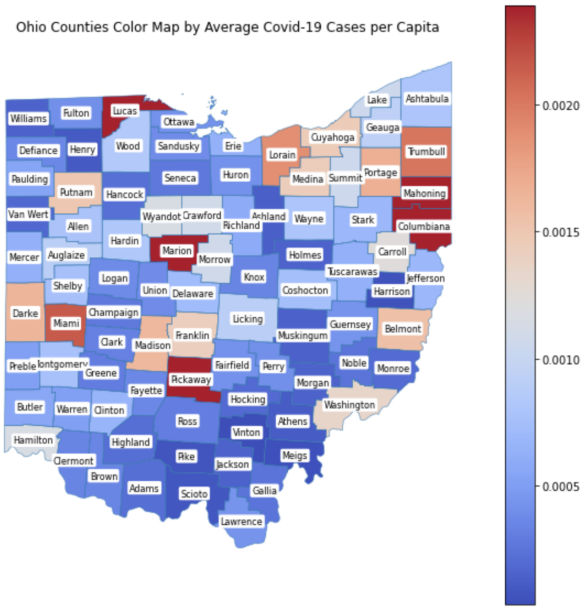

Fig. 3. Color Map Average Deaths per Capita

Further geographical analysis(Fig. 2 & 3) at the county level revealed Delaware County as having the highest average core Jaccard normalized awareness value, indicating possible variances in information dissemination or public engagement across regions. The dataset also allowed for the examination of COVID-19's impact relative to population, identifying counties like Pickaway and Marion with higher cases per capita, and Miami and Darke with higher deaths per capita.

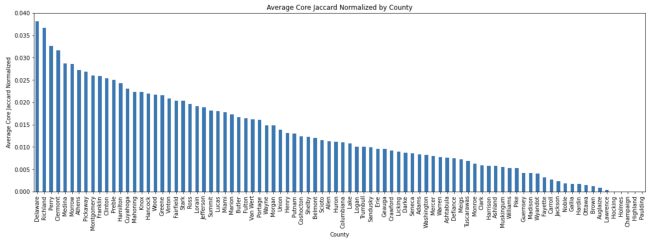

Fig. 4. Average Core Jaccard Normalized by County

As in Fig. 4, the high level of awareness in Delaware County may reflect successful local public health campaigns or a community that is more engaged with health information. On the other hand, the variation in COVID-19 cases and deaths per capita across counties like Pickaway and Marion versus Miami and Darke suggests differing levels of disease spread and mortality, which could be attributed to factors like population density, healthcare access, and local mitigation efforts.

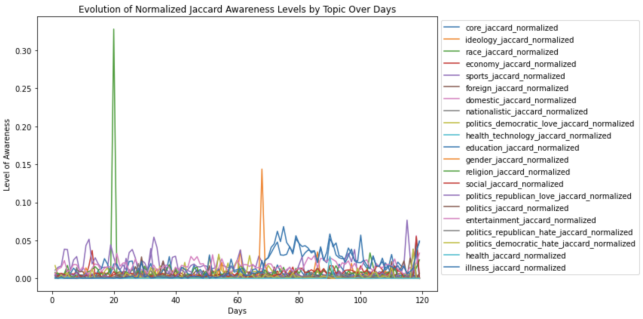

Fig. 5. Awareness Levels by Topic Over Days

Temporal analysis on topic awareness over days illustrated fluctuating levels with no single topic consistently dominating public attention, suggesting dynamic changes influenced by various external factors. The trend showing the ebb and flow of various topic awareness levels over time, highlights the dynamic nature of public interest and concern. These fluctuations can provide valuable feedback for policymakers and public health officials on when and how certain messages resonate with the public or when they need to ramp up communication efforts in response to emerging issues or waning attention.

III. METHODS

A. Data Prepossessing

The first step in any kind of ML model training is to get the set of pre-processed data frame. We have filtered out all non-numeric columns and used only numeric columns for training our model. We then create an output data frame by reading the sample output file provided to us. This step could have also been done without reading the file, but it is always a good practice to use a file as a standard output template (in case it changes in future, you only need to change the output template file instead of changing the code base and re-running the entire thing)

We have modified the categorical variable of "county" as integers that we will be using further on for our analysis. We have defined a set(or a list in this case) of mandatory columns that we feel are mandatory for analysis From the selected numerical columns we are then extracting a set of mandatory columns for both testing and training data respectively.feel are essential for the machine learning model. It contains columns like total_pop, deaths, percent_highschool etc. The list mandatory_cols_for_test is the exact same list as above with the exception of the "cases" columns as that is the one that we need to predict. The lists normalized_cols and un_normalized_cols are self-explanatory, they are a list of columns that are normalised and columns that are not.

Then we do have lists that also store jaccard_cols cosine_cols and intersection_cols. There columns can be used to reporpose the data for training. We can remove specifically jaccard and intersection columns and work only with cosine columns if needed. As of this moment since we already filtered out the columns earlier these lists will be blank but can be used for future improvements.

B. Model Setup

We have then a code to print out the first row correlation matrix, in order to cross verify the linear co relation between our target variable and other columns in our finalized data set. Following this, we have our logic to train our finalized dataset. We are using RandomForestClassifier for this as Random Forest combines the forecasts of several decision trees, it is resilient to noisy data which is a necessity in this dataset.

We have commented out train_test_split since that was being used for internal testing to verify accuracy, r2_score and mse(Mean Squared Error). We are then appending our results to output_dataframe and placing it in a csv file.

C. Verifying the strength of the trained model

To get the best value for n_estimators we have run the RandomForestClassifier over a range of values from 100 to 300 with the same random state for improved precision. We have then plotted a graph to analize r2_socre against every n_estimator and picked the best fit. An important thing to note here is we decided not to go with values less than 100 as it might give a better r2_score however the results were highly inaccurate. We are then printing out the max_ids which is the best fit for n_estimator. Make sure to scale the value accordingly, example if we decide to loop over 100 to 300 and the max_ids is 38, then the best estimator value in 100 + 38 (since max_ids here is the index and not the value itself.)

IV. RESULTS

By eliminating most of the columns and just by using the mandatory columns we did achieve a high r2_score of 0.955 and mean squared error of 33265.001. The accuracy was a bit on the lower side of 0.664 but it is important to note that this was verified using train_test_split the accuracy after uploading the results against the testing data set was around 0.89.

What we could have done better is to use other columns apart from the ones mandatory to improve accuracy, not all the columns were required, some columns that actually made sense were excluded from the data set(example core_cosine_normalized)

REFERENCES

[1] GeeksForGeeks, Random Forest Regression in Python, https://www.geeksforgeeks.org/random-forest-regression-in-python/

[2] StackExchange: Find the optimal n_estimator by looping, https://datascience.stackexchange.com/questions/69930/find-the-optimal-n-estimator-by-looping-the-model-accuracy-indicator-in-random-f

[3] Wikipedia. Link extracted from lecture slides and assignment pdf, https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Ohio