The Impact of Environment on COVID-19

Kristina Mishra
Jay Ghodke
Neil Coffman
Nathan Leon

1. Introduction

Of the tasks available in the COVID-19 dataset, our team decided to participate on the ‘What is known about transmission, incubation, and environmental stability’ task. Specifically, we wanted to focus on the role of the environment in transmission. To that end, we will attempt to build a model that will identify the most relevant documents covering the topic. By building this model we will be able give the best representation of the data to determine what criteria will determine transmission levels.

Identifying relevant documents and scoring them incorporates using a page ranking methodology that is based on predefined specific terms. A dictionary with environment-rich terms was created as a basis to rank each document. TF-IDF was used against the title and abstract for terms that match the dictionary. Normalization was then applied against each TF-IDF value followed by a final ranking that summed up all final values for each document.

2. The Dataset – Assumptions, Limitations, and Implications

Of all the data provided with this Kaggle challenge, our team felt that we needed to use the metadata.csv data source for the entirety of our analysis. With 57373 records, the dataset was assembled by the Allen Institute for AI for Kaggle to include high level overviews for a variety of scholarly articles related to coronaviruses. Most notably, the title and abstract fields contain the most detail about this list of papers, all of which is available within the metadata.csv file.

A limitation of using just the metadata file is that we did not have access to the full document text. While some publications had their full text provided by Kaggle in a parsable format, the computational power to apply our analysis approach to tens of thousands of research articles would have been prohibitive.

Our approach focused on seeking out English words when establishing usefulness scores. This restricts the usefulness of our analysis for the many articles in the dataset not written in English. These articles could contain valuable information to better create a model and understand our problem statement. Although this is a limitation of the current iteration of the project, further features could be added to include language in the future.

The text filtering and analysis was done with the assumption that the articles collected by Kaggle were associated with coronaviruses or respiratory issues that would be noteworthy for researching COVID-19. As a result of this assumption, we did not apply any extra filtering or scoring based on how well it aligned to COVID-19. There are also some issues with data completeness, and while all papers have been listed with a title, 10519 – around 18% do not have the abstract filled out. Throughout this paper we will address different methods on how to deal with this problem.

3. Initial Data Processing

With the limited scope of our effort, there are numerous columns in the provided metadata that are not useful for our efforts and can be dropped prior to other operations to reduce memory impact. The surviving columns after this update were: cord_uid, title, pubmed_id, abstract, and journal.

Rogue characters and inconsistent capitalization could hamper our text detection approaches, so as another pre-processing step, all words of the title and abstract text fields were shifted to containing just lower-case letters to make the data more uniform. To remove garbled text likely caused Kaggle’s webscraper, all non ascii characters were removed. An example of this cleanup would be the interpretation of" â€œsuspected caseâ€ and â€œconfirmed caseâ€” to “suspected case and confirmed case”.

Once these initial pre-processing steps were completed, the title and abstract were incorporated into a corpus for additional processing. Data processing of the corpus included removing punctuation, removing [English] stopwords, stripping white space, and removal of additional characters not included in the previous processing. Finally, the corpus terms were stemmed to their root to ensure that different forms of the same words would be counted as one term and eliminating potential scoring anomalies that would otherwise occur if stemming were not performed.

4. Text Mining

4.1 Identifying Key Environmental Words

Our team worked together to create a set of words that we felt were associated with environmental phrases and intents. This set was targeted to include words for both indoor and outdoor environmental conditions and was sourced using both domain knowledge as well as standardized glossaries for weather and environment such as https://w1.weather.gov/glossary/. Some examples of these are ‘heat, HVAC, seasons, humidity’. In all, 116 words were included in the final set with the ability to add more as needed. The final set resulted in a dictionary text file to use as the base to score our documents against. The dictionary was stemmed to ensure accuracy when matching to terms in the corpus.

4.2 Document Scoring

Document scoring provided a unique challenge given the use of singular terms provided from our dictionary rather than multiple word queries. Topic modeling was considered, but given we were only initially identifying one topic and the complexity involved, this was determined to be out of the scope for this project. As such we employed a bag-of-words vector space model using a simplified informational retrieval methodology.

The first step was to create a document term matrix to represent each term from the title and abstract across all documents for terms appearing in the environment dictionary. Our document term matrix values were weighted by term frequency (tf) and inverse document frequency (idf), given a higher weight to the terms that occurred more in any given document but that rarely occurred in all other documents. This would allow us to include important terms, but also give weight to potential novel information. The assumption being we would want to give weight for important terms such as environment, but also give an additional boost for more unique terms such as pollen and precipit (the stemmed version of precipitation).

To compensate for variations in document length (including missing abstracts), the document term matrix was normalized by using the following formula: $\sqrt{\Sigma \mathrm{x}(vector\ matrix^2)}$ (where x = a term value for every document). Each resulting value was then sum totaled across a document to give it a final rank score.

5. Regression Model

5.1 Model and Feature Selection

Text mining or text categorization poses exceptional challenges due to the very high dimensionality of text data, sparsity, multi-class labels and unbalanced classes. In addition to this, digitalization of various processes is causing drastic increase in volume of text data from internet, databases, and archives. Many predictive modelling methods have been developed for analyzing text documents, such as random forests, support vectors machines (SVM), K-nearest neighbors, and decision trees.

Review of the project task objective indicated a need to change the modeling method from a classification model to a regression model[1]. Adding a class feature with values: likely/not likely was considered, but it was determined that the ranking of the instances provided more useful information. Choosing an arbitrary number such as .5 would mean that documents with a score of .49 would rank the same as a document of .1 instead of showing the closeness to documents with a ranking of .51. Regression aligns better with a continuous response and thus preserves important information, instead of filtering down to abstract buckets that would result from using classification. In both predictive modelling methods, regression predicts a continuous value and classification predicts the belonging to a particular class. Classification is preferred over regression when the results of the model need to return the belongingness to specific categories. Since we wanted to find out the correlation of the information provided in each document to environment terms, we chose regression to get the relevance score.

Random forest is a supervised learning algorithm which uses ensemble learning method for regression. Due to its algorithmic ease and prominent regression performance for high dimensional data, random forest has become a promising method for text analysis. The ability to handle large databases and thousands of input variable makes it a robust and highly accurate model. The random forest enhances accuracy through randomization and averaging of the trees by decreasing correlation without drastically decreasing strength. When a particular tree overfits on a particular training set, other trees based on other training data are unlikely to overfit in the same way. Similarly, when a randomized tree overfits on a training set, a different realization may not do so on the same training set. The algorithm creates more randomness when growing trees and it searches for the best feature among an arbitrary subset of features instead of searching for the best feature while splitting the node thus, yielding an overall better model. Random forest provides superior bias-variance trade-off due to it's already low bias and the ability to reduce variance with parameter tuning.

Two additional model types were considered: support vector models (SVM) and kNN. SVM was rejected as it tends not to perform well with large data sets and did not have a favorable trade off for complexity of model. Similarly, kNN was rejected as it is computationally expensive and is

[1] As per our discussion with Dr. Olafsson

sensitive to outliers. Both algorithms would have required additional processing time for multiple testing of feature selection that was performed automatically with random forest.

One of the problems with the randomForest package in R is its single threaded nature. An alternative approach we found was to use the ranger package instead. Ranger employs a randomForest algorithm but supports multi-threading – allowing our models to be generated in a small fraction of the time and allowing us to explore feature selection without many hours of modeling time.

In hyperparameter tuning of the models, we iterated across 4 attributes: mtry, splitrule, min.node.size, and numtrees.

mtry – Number of features selected for each bagging iteration of the algorithm, tried for (1/10, 1/5, 1/4, 1/3, 2/5, and 1/2) of the feature set

splitrule – Approach to how the nodes in the tree are built, tried for (variance, maxstat, extratrees) These 3 approaches are recommended for use with regression.

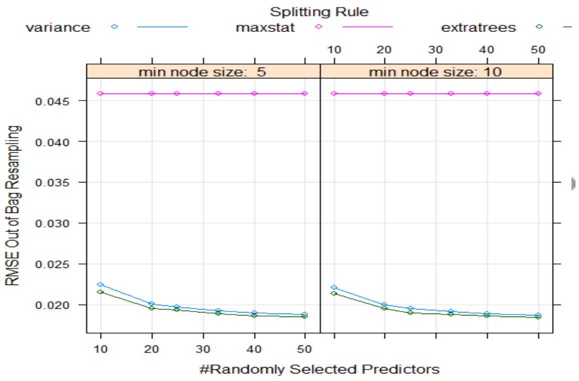
min.node.size – Number of records at which, the which the algorithm will not build a new level of the tree, tried for (5, 10) For regression type modeling, ranger recommends min.node.size of 5.

num.trees – Number of trees to be built by the algorithm, generally more is better, but computationally expensive, tried for (200, 500, 1000, 1500)

Feature selection was tested on two levels. Initially this was done using Boruta, but it did not reduce any features other than features we had already identified for removal due to sum zero values. It did give us some interesting metrics for feature importance, but it was computationally expensive often taking over 3 hours to run. Since we were already using a random forest model which employs feature selection via subspace sampling, there was no additional need to evaluate the importance of features that ended up different than our results of importance of features for random forest that is shown in Figure 6 below.

5.2 Diagnostics

Looking at the initial results of the modeling output we can immediately see that maxstat, despite being recommended for regression, simply does not perform well. All maxstat models can instantly be discarded from consideration. Variance also performs below extratrees, so it can be removed.



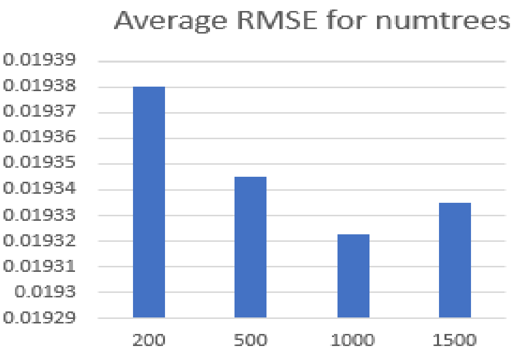


Figure 1: Tuning results from num.trees=1000 *Figure 2: Average RMSE for selected numtrees*

Across all models, models with 1000 trees tend to perform better for RMSE with less mtry values (Table 1). R^2 values are slightly lower, but are of less of a concern since the mtry is lower and R^2 can have an artificial increase as more predictors are added to the model. A larger mtry value gives more bias to the feature selection, so that is further evidence that the 1000 tree model is the best selection for additional parameter tuning. Further analysis was done with num.trees set to 1000.

	200/mtry50	500/mtry50	1000/mtry33
RMSE	0.01857	0.1845	0.0159
Rsquared	0.858	0.8602	0.8321

Table 1: Metrics evaluation best tuning for different num trees

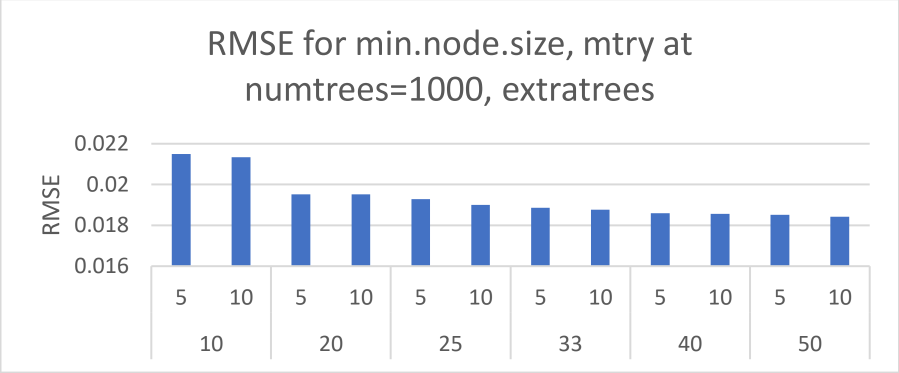


Figure 3: Remaining model evaluation

As mtry reaches higher values we risk overfitting our model, although this is less of a concern with random forest due to its bagging capability which lowers variance. To be safe, an mtry of 1/3 the the data set size was selected, since a lower mtry gives greater variance with feature selection with each sample and thus less bias to specific features – although it does still retain some bias for features with more values. This gets most of the performance improvements from the lower ratio values.

The min.node.size appears to not affect performance that much, so we will differ to the suggested regression value of 5.

A test for normal distribution of the residuals was performed (Figure 4), showing a generally straight line. Both the right and left side tails have a skew giving an indication of potential outliers, but most of the data points appear to not violate the normality assumption.

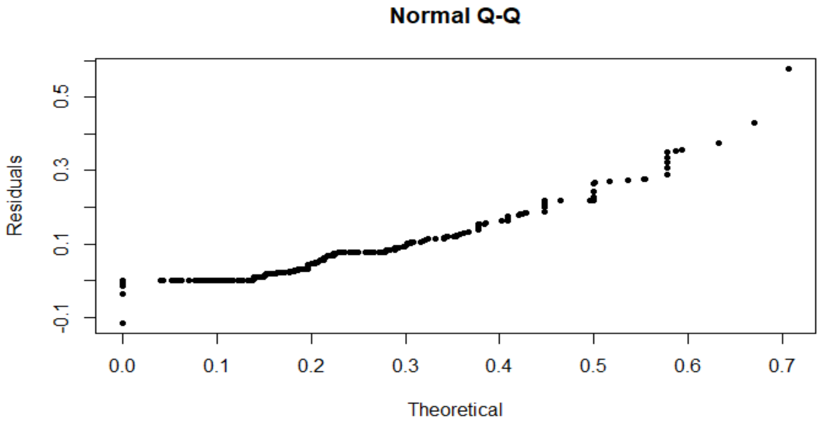


Figure 4: Normal Q-Q plot of prediction residuals

6. Results

The final RMSE of our predictions was 0.01597078. Additional metrics can be seen below:

Type:	Regression
Number of trees:	1000
Number of independent variables:	100
Mtry:	33
Target node size:	5
Variable importance mode:	permutation
Splitrule:	extratrees
OOB prediction error (MSE):	0.0003521017
R squared (OOB):	0.8321982

Table 2: Attributes and statistics of selected ranger model

When plotting the residuals across the range of our scored values, a trend emerges showing that the higher ranked papers tend to have much higher residuals. This is likely a result of very few ‘good’ environmental papers in the data set, making for limited training opportunities.

While modeling, attribute importance was documented. Plotting the importance shows a slow decline after the first 20 or so attributes.

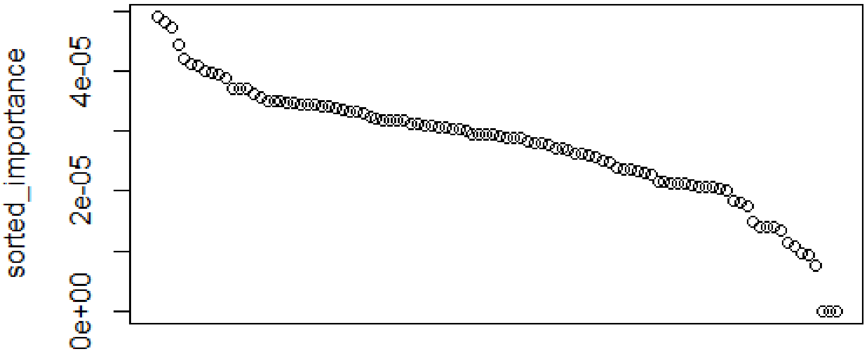


Figure 5: Importance across all attributes

Viewing the top 25 of these shows the words that it found most useful across our attribute list:

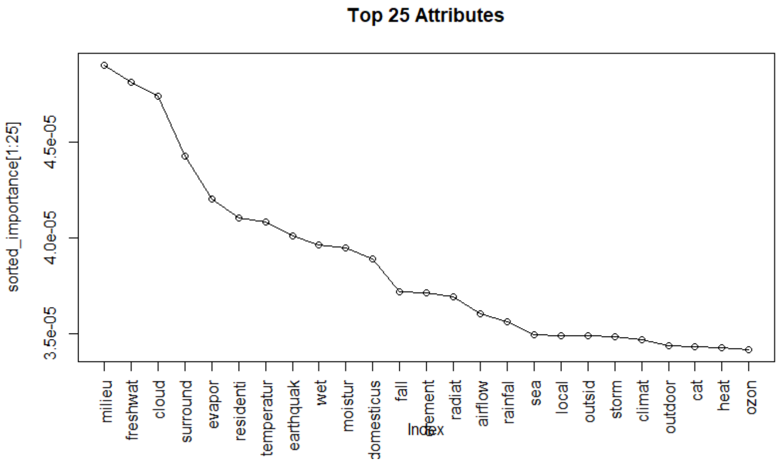


Figure 6: Importance and names of the top 25 attributes

7. Recommendations

Based on our results, the optimal model for random forest (using ranger) uses the following paramenters: splitrule = extratrees, mtry = 33, mid.node.size = 5, and num.trees =1000. No additional feature selection was required in addition to what was determined by random forest, but feature correlation should be considered with lower performing features observed in Figure 6 (e.g. outside and outdoor); though still with caution. Any feature selection performed outside of random forest would likely change the results of the tuning parameters.

The model produced through our analysis consistently underpredicts the likely score of the article when it is a non-zero score. Our ability to correctly identify 0 based scores (and the large quantity of 0 scoring documents in our set) is what gives us such a good RMSE. When using our model to flag potentially useful environmental papers, we recommend setting a threshold of 0.15, and reviewing by hand any articles above that value. In our test set, 0.15 removed 98.7% of articles from pool while only falsely eliminating one well-scored paper.

8. Conclusion

COVID-19 has come into our lives with little warning and at the current moment we know very little about it compared to other infectious diseases throughout history. While this project only touched on one aspect of environment, the research conducted here could still be useful to the research community. With our model, we were able to turn 100+ words that have some connection to our project statement of ‘What is known about transmission, incubation, and environmental stability’ into a matrix and set of words that would best describe the dataset.

The approach used for this project allows for multiple expansions including considerations for additional attribute construction, topic modeling and query ranking. Some text mining improvements that could be made would be adding an attribute, perhaps a 0-1 scalar, to indicate if an article focused on viruses. By implementing topic modeling techniques, multiple topics (e.g. seasonality, virus shedding, persistence on surfaces, etc..) could assist researchers across many domains. Adding query ranking capabilities and implementing cosine similarity into the scoring of the documents would further enhance its usefulness. Multiple dictionaries in multiple languages (including applying language capabilities across the data engineering tasks) would create a robust search engine that is specific to COVID-19. And finally, additional considerations such as mining the entire document and weighting the various sections by importance could improve accuracy, though was computationally prohibitive for this project.