**Challenge 1**

**Not The Corona Virus Team Member’s Information**

|  |  |  |
| --- | --- | --- |
| **Team Member Name** | **Email Address** | **Challenge** |
| Cesar Diaz | [Calexdiaz143@gmail.com](mailto:Calexdiaz143@gmail.com) | 1 |
| Rhodora Villanueva | [Rhodora33.rv@gmail.com](mailto:Rhodora33.rv@gmail.com) | 1 |
| Jaimie Gillette | [Jaimie.gillette@gmail.com](mailto:Jaimie.gillette@gmail.com) | 2 |
| Timmy Ngo | [Timmyngo021@gmail.com](mailto:Timmyngo021@gmail.com) | 2 |
| Randy Rodriguez | [Randy.renezz@gmail.com](mailto:Randy.renezz@gmail.com) | 3 |
| Carlos Sanchez | [Sanchezcj95@gmail.com](mailto:Sanchezcj95@gmail.com) | 3 |
| Annie Voigt | [a.voigt1121@gmail.com](mailto:a.voigt1121@gmail.com) | 3 |

**Overview of Approach**

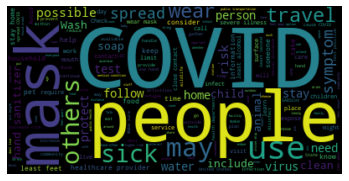
Our approach for Challenge 1 was to use some classic Natural Language Processing (NLP) solutions and 1 homegrown algorithm to get a grip on the semantics and danger levels of each scenario. We used sci-kit learn for our bag-of-words vectorizing approach, and gensim for our word embeddings approach. We also used Natural Language Toolkit (NLTK) throughout to help us with parsing the dataset. As for training/deciding on our models, we went for an ensemble of several popular models (Naive Bayes, Linear Regression, KNeighbors, SVC).

**Technical Description of Solution**

We had 3 separate approaches to this problem that were all combined in our final ensemble predictor.

For our first approach, we did a bag-of-words Term Frequency-Inverse Document Frequency (TFIDF) vectorization of the dataset with n-grams of sizes 1 to 3. We prepared the dataset by firstly lemmatizing the string values using NLTK’s WordNetLemmatizer. We decided not to exclude stopwords after some run-throughs revealed a positive impact on performance (possibly due to negative stopwords such as “won’t”, “weren’t”, etc.). We then modeled the modified dataset using Linear Regression, Kneighbors, and SVC training methods. To tune our hyperparameters, we used SKlearn’s gridsearch package to test some of the different loadouts for each model.

Our second approach was probably our most creative. For this solution we chose to use word embeddings but not in the standard average-embedding approach. Firstly, we converted every word in our dataset to their embedded equivalents if available. Then, we congregated a list of “hot topic” words that we thought would be extremely relevant in predicting on our dataset. We did this by developing word-bubbles of the CDC Guidelines available online.



We chose relevant terms from this list such as “mask,” “coronavirus,” “travel,” and “home.” We also added some words of our own that exhibited significant improvements when testing (“asthma”, “outside”). Then, we created a new dataset. This dataset would be comprised of the minimum distances found between each relevant term and word embeddings in our original scenarios. This new dataset encoding each scenario’s closeness to the valued words was now what we ran through our ML algorithms. As for the algorithms chosen we stayed with the same approaches as problem 1 (Linear Regression and SVC).

Our third approach we used was the “Average Embedding” solution. This involved us turning each word into its embedding equivalent. Then we took the average of all embeddings in a row and used that average as our value for each row. Once we finished this, we ran this through LinearRegression and SVC models.

Finally, we did an ensemble majority vote across all of the models we trained from our 3 approaches and this ended up being our final predictor for danger\_level.

**Reason(s) for Choosing the Approach Taken**

Our approach attempted to use 2 industry-standard ideas (Bag-of-Words and Averaged Embeddings) for accuracy and 1 homegrown solution for creativity in this competition.

This approach should meaningfully scale well with larger datasets and possibly even perform better given more labeled data. There was a high variance in our testing results due to the small dataset size. So with a larger dataset, it might also be easier to get the exact right tunings on our hyperparameters as well.

In terms of how our approach would generalize to new data, given a new pandemic or natural disaster we’d be able to adjust our relevant terms list in Approach#2 to fit those scenarios more specifically, and our Bag-of-Words and Avg-Embedding solutions would work in their current state.