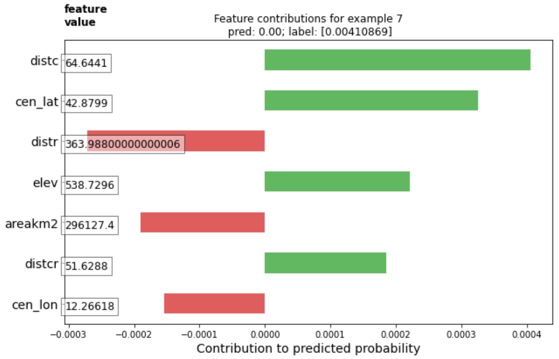
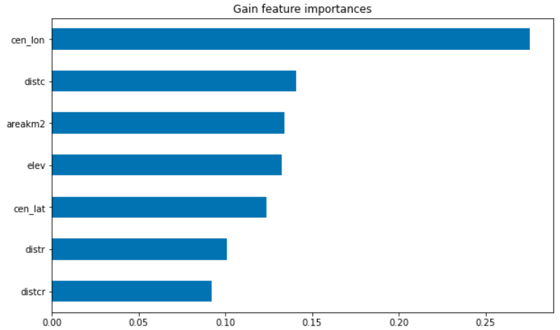
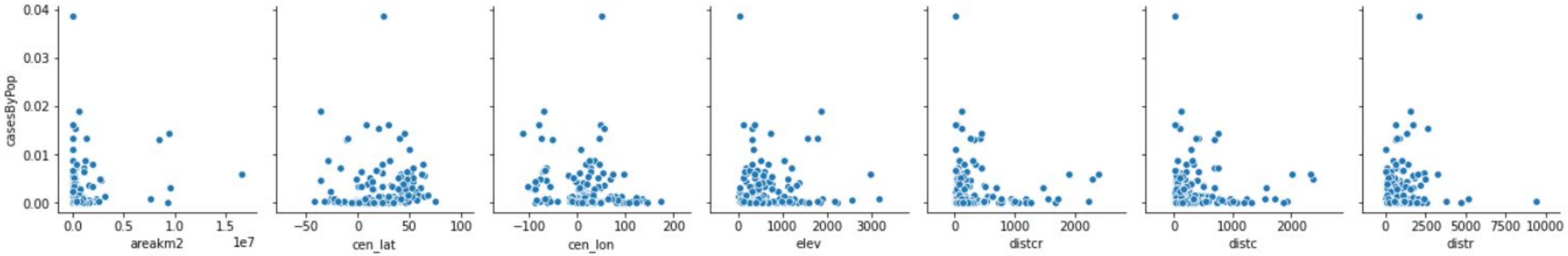
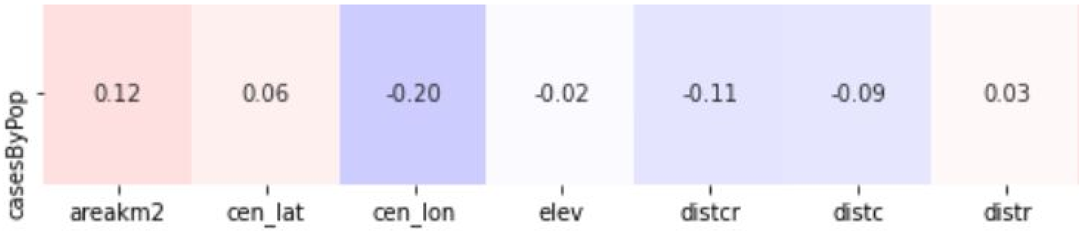
Right now, every country has their own data for how many people have confirmed cases of COVID-19. As time goes on, this number will inevitably (through logic), level off at some point. This model will be able to work with more accurate data as time goes on, but is there a way we can be able to determine the percentage of persons affected with COVID-19 within a country based on that country's geographical location (relative to the equator), (average) elevation, size (land), and position (distance to major bodies of water, such as oceans)? By doing so, we may be able to find out more about the disease and how it spreads (such as the disease being attracted to humid places near bodies of water, thus affecting countries like Mexico more than "landlocked" countries like Switzerland and Hungary). There will be outside factors that contribute, such as the country's regulations for masks for prevention, and other differences such as culture or access to medical preventative services.

The data being used is actually from 3 different sources. I needed to get coronavirus data, country data (numbers such as population), as well as the country’s geographical location data (such as elevation). This uses a combination of datasets that must be filtered through each country’s iso codes, as those are always 3 letters and unified (this avoids country formatting mismatches between datasets, such as United States vs. U.S., etc). The sources are from Google; from Our World in Data and The College of Urban Affairs.









The method used was using the BoostedTreeClassifier. I initially wanted to do this because not only was it the most recent one, but it also appealed to me immediately since I wanted to show off all of the different dimension in the data. I thought that boosted tree would also be good because I know that trees will decide to split based on the biggest noticeable split in the data at each node, so I thought that being able to differentiate whichever one of the many possible dimensions would be good for this. Instead, I failed to realize that until I finished the entire project, that the reason why the accuracy was so low (0.0) was because the boosted tree classifier works best with more categorical data, where getting the label would be a category, and not continuous.

There was not a good way to assess the model’s accuracy. Though, the loss was decently low for the boosted tree model. Even if I did do a linear regression model, the results wouldn’t have been too significant (as shown), either, as continuous data is literally, “continuous,” and thus can have its data directly shown upon the determination (label) variable on a chart. Even if ideal, it still has a difficult time being able to find strong correlations to the result.