Machine Learning Engineer Nanodegree

Capstone Project

Christopher Weaver

August 26st, 2021

I. Definition

Project Overview

The past couple of years have presented us with a unique opportunity to utilize increasingly accessible hardware and sophisticated software to enhance individual health outcomes and knowledge. Smart phones continue to increase, not only in computational power, but also in the array of sensors these devices contain. With the release of the Apple Watch Series One, Apple made a marketing and engineering commitment to empowering users to monitor their own health through technology. In recent years, wearables such as the Apple Watch have been fitted with sensors to help users monitor heart rate, blood oxygen levels, and even check for cardiac arrhythmia. As these sensors become more prevalent, it opens up opportunities for software engineers to find new ways to use these sensors. Monitoring heart disease, physical activity, and environments that can lead to hearing loss have already taken shape; but there is more that can be done. Companies such as Cardiogram are banking on just that. Research conducted in partnership with the University of California found that utilizing LSTM neural networks and “off the shelf” wearable heart rate sensors, medical conditions such as diabetes, high cholesterol, high blood pressure, and sleep apnea could all be detected with high accuracy[1]. I propose that similar techniques can be extended beyond just these diseases into a whole host of other ailments and conditions.

Problem Statement

In the spring of 2020, individual cities, counties, and states began implementing restrictive mitigation strategies around the novel corona virus SARS-C0V-2. Many feared either contracting or spreading of the virus and took to quarantining at home for

[1] AAAI Conference on Artificial Intelligence, Feb 2018 (AAAI-18). https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16967/15916

weeks or months at a time. Many institutions scrambled to find ways to resume semi-normal activity while at the same time keeping all involved safe and comfortable. Different testing solutions were rolled out as a mechanism for helping individuals know if they were infected with the virus so that they could make better informed decisions. But many of these tests are slow, expensive, inconvenient, and too inaccurate for individuals to take on a regular basis. What is needed is an easier way to test and monitor for Covid-19 that empowers more people to make responsible health decisions.

The goal is is to build a classification machine learning model that can accurately diagnose Covid-19 using bio-metrics gathered either by api's to Apple's Health app or from user input to specific questions such as "Are you currently experiencing a cough?". Inputs would include numeric values (body temperature, pulse, and Sp02) to output a class output of either 1 or 0, with 1 representing a Covid-19 diagnosis and 0 representing no indication of infection with Covid-19. A further model would be developed which takes binary classification values such as the existence of a cough, sore throat, and headache and again outputs a class value of 1 or 0, with 1 representing a Covid-19 diagnosis and 0 representing no indication of infection with Covid-19. Each feature in the second model will be represented as a 1 if the user is experiencing the symptom or a 0 if they are not.

Metrics

The most common test for Covid-19 is a PCR test, usually done by a nasal swab. Some studies suggest that the sensitivity of a PCR test may be as low as 72% for those displaying symptoms and only 58.1% for those not[2]. False positives are almost non-existent for PCR tests which means they have an extremely high specificity. Since our solution involves using measurements of user symptoms to diagnose and predict Covid-19, we will only benchmark against PCR tests when displaying symptoms.

Standard evaluation metrics can be used for our solution. The two we will most focus on are sensitivity and specificity. When it comes to Covid-19, we much prefer a user gets a false positive than a false negative. False negatives will lead to users potentially acting in a manner more likely to infect others which goes against the major point of this project. Standard train test splits will be used to help us evaluate our models, with particular attention paid to both sensitivity and specificity when testing against the test set. For our second model when considering against multiple classes, we will evaluate the precision and recall/sensitivity for our models.

Because we are attempting to find better and easier ways to help people identify if they are infected with Covid-19, common tests such as PCR are a great benchmark to try and beat. Realistically, because our solution to the problem is significantly cheaper

---

[2] https://www.healthline.com/health/how-accurate-are-rapid-covid-tests#how-accurate-is-it

(assuming the user already owns an iPhone and Apple Watch) and far more convenient, we would probably be justified in having a lower sensitivity target than the PCR test.

II. Analysis

Data Exploration

I will be utilizing two different datasets to help tackle the issue. The first dataset contains biometric data usually referred to as vitals, such as temperature, pulse rate, and Sp02 levels[3]. Each row of the dataset contains one measurement for each and a binary classification column for either currently being infected with Covid-19 or not at the time of the measurement. Sp02, Pulse, and Temperature will be my features and the Result will be my target. All three features are continuous values and will need to be normalized in order to ensure any model we use does not think one feature is more important than any other. My target is a string which will need to be changed to a binary target of 1 and 0.

	ID	Oxy	Pulse	Temp	Result
0	0	98	65	95	Negative
1	1	96	92	95	Negative
2	2	95	92	99	Negative
3	3	97	56	96	Negative
4	4	88	94	98	Positive

The second dataset to be utilized contains many more biometrics in binary classification form. These include cough, muscle aches, tiredness, sore throat, among many others[4]. Each value is represented as a 1 if the user had the symptom and a 0 if they did not. Each row also contains a value for diagnostics which could be Flu, allergy, cold, or Covid-19.

DIFFICULTY_BREATHING	LOSS_OF_TASTE	LOSS_OF_SMELL	ITCHY_NOSE	ITCHY_EYES	ITCHY_MOUTH	ITCHY_INNER_EAR	SNEEZING	PINK_EYE	TYPE
0	1	0	1	0	0	1	0	1	ALLERGY
0	1	0	1	0	1	1	1	1	ALLERGY
0	1	0	1	0	0	0	0	1	ALLERGY
0	1	1	0	0	1	0	1	1	ALLERGY
0	1	1	0	1	0	1	1	1	ALLERGY

[3] https://www.kaggle.com/rishanmascarenhas/covid19-temperatureoxygenpulse-rate

[4] https://www.kaggle.com/walterconway/covid-flu-cold-symptoms

Preliminary investigation into this dataset shows that it does not possess an even distribution among the possible target values. The Flu is significantly higher than any other target. The Flu and Allergy make up more than 90% of all values for our target. Our unbalanced dataset could result in a model that looks very accurate, but does poorly when it comes to predicting Covid-19 as the model found learning about this target not relevant to obtaining a low error rate.

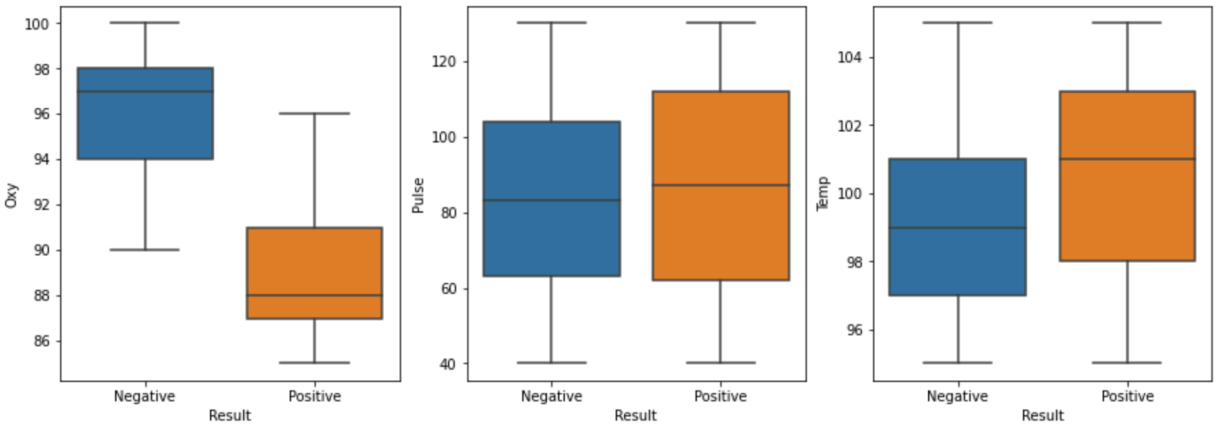
```
FLU        25000
ALLERGY    16381
COVID       2048
COLD        1024
Name: TYPE, dtype: int64
```

Exploratory Visualization

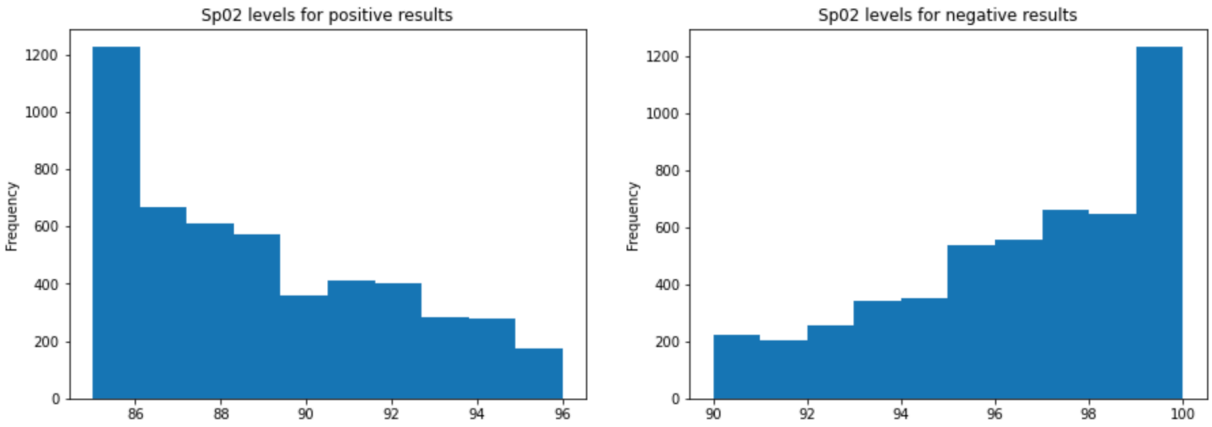
Initial analysis showed to my surprise little correlation between different vitals on our first dataset. I would have expected some correlation between Sp02, Pulse, and Temperature. Despite this lack of a relationship, we do find a high correlation between Sp02 and positive tests for Covid-19. We also see a modest relationship between temperature and positive tests for Covid-19.

	ID	Oxy	Pulse	Temp	Result
ID	1.000000	-0.002693	-0.012499	-0.009279	0.011395
Oxy	-0.002693	1.000000	-0.005724	-0.015681	-0.777019
Pulse	-0.012499	-0.005724	1.000000	0.009602	0.050590
Temp	-0.009279	-0.015681	0.009602	1.000000	0.262522
Result	0.011395	-0.777019	0.050590	0.262522	1.000000

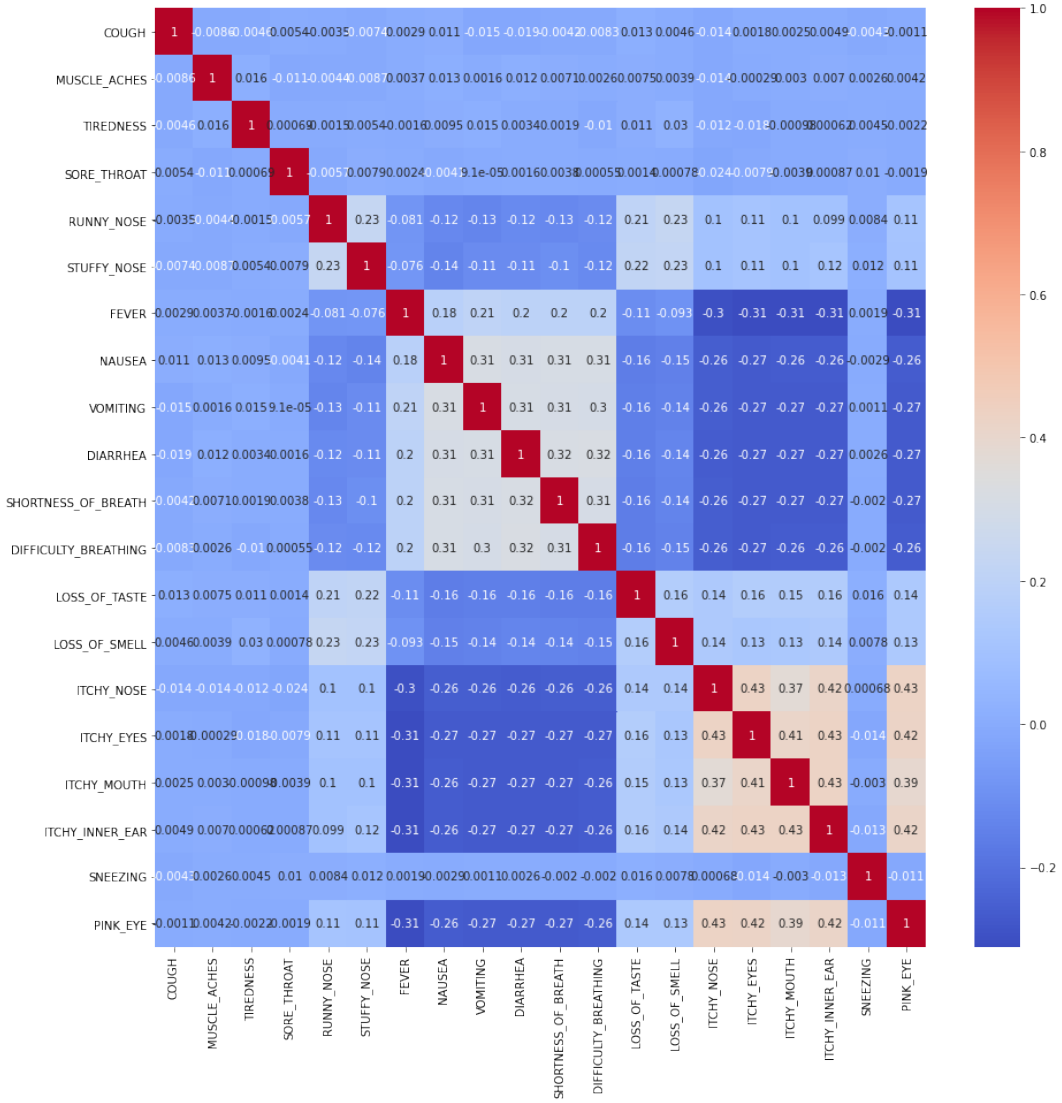
We can see that the largest difference between positive and negative Covid-19 cases is found for Sp02 features. This corroborates the correlation values we see above. There is a noticeable difference in the mean for temperature, but quartiles are relatively close together. Pulse looks nearly identical.



Looking at distributions for Sp02 for both positive and negative Covid-19 cases shows values skew towards extremes but no definitive demarcation line exists.



For the second dataset, I built out a heat map for feature correlation to help identify anything of interest. This was helpful since we have 20 features to consider.



We see some correlations in the second dataset between different itchy body parts as well as some weak correlation between Fever, Nausea, Vomiting, and Diarrhea. Nothing significant can be found though.

Algorithms and Techniques

Our problem is best solved by developing two different models, one for each dataset. For the first dataset, I will focus on training two different models that I think might work well for our use case. The first is a decision tree. Because we are not dealing with a complicated feature set and because Sp02 showed decent variability between positive and negative Covid-19 cases, I thought a decision tree might be a good place to start. One of the advantages of the decision tree is its easy to understand how the model came to its conclusion. We can evaluate the model itself by graphing it out and gaining further insights into how Covid-19 presents itself in those infected with it. I also will train a logistic regression model as a secondary model. Logistic regressions are very simple and work for our case since we are only looking for a binary classification of positive or negative for Covid-19. There are two advantages of this model over a decision tree. The first is that it is less likely to overfit the data. The second benefit is the logistic regression model can output a predictive probability for both positive and negative outcomes which might be helpful for us later. I may want to present to the user the confidence level of a negative or positive diagnosis from the model.

Part of the purpose of this project is to design an iOS mobile app that can monitor Covid-19 without much needed from the end user. An Apple Watch series 6 can monitor pulse and Sp02 levels throughout the day, but not temperature. Temperature readings would require user input to achieve. I will also train both the models above with only the Sp02 and Pulse data and see how well they do. If the results are good enough, I might forgo using temperature as a feature.

Our second dataset has many more binary classification features to work with, but also has four classes to predict between as an output. The extra dimensions we have in this dataset over the first made me decide to use a random forest as my first model over a decision tree. I found using 5 estimators produced the best outcomes. I also will train on a support vector machine. I suspect a svm might work well since they tend to be more effective in high dimensional spaces. Since I need to model to predict between four classes, I will probably need a kernel that is non-linear. I will experiment with both an rbf and a polynomial kernel.

Benchmark

Both models will be doing classifications on the features they are fed. I will be paying attention to both the precision and specificity of the outputs of the models. Priority will be for a very high specificity. False positives are a worse case scenario when informing users about potentially having an infectious disease. I will likely benchmark the results

of my models against what we generally see for PCR tests. An accuracy above 70% is probably sufficient so long as the specificity is above 90%

III. Methodology

Data Preprocessing

For our first dataset with vitals, I need to convert the targets from TEXT to integers. I will covert all ‘Negative’ to 0 and ‘Positive’ to 1.

```
originalData_df['Result'] = [1 if x == 'Positive' else 0 for x in originalData_df['Result']]
originalData_df.head()
```

	ID	Oxy	Pulse	Temp	Result
0	0	98	65	95	0
1	1	96	92	95	0
2	2	95	92	99	0
3	3	97	56	96	0
4	4	88	94	98	1

I next need to take the dataset and separate the features from the targets. The features are then to be run through an Scikit-Learn Standard Scaler in order to normalize the data for training.

For the second dataset, preliminary investigation into this dataset shows that it does not possess an even distribution among the possible target values. The Flu is significantly higher than any other target. The Flu and Allergy make up more than 90% of all values for our target. I originally ended up choosing 1024 records from each Type for training but found that my models found differentiation between Covid-19 and the Flu very difficult. So I changed this initial modification to 2048 records per case (except for the cold which remained at 1024). This gave my models the data it needed to better handle Covid-19 and Flu cases.

```
def sampling_k_elements(group, k=2048):
    if len(group) < k:
        return group
    return group.sample(k)

originalData_df = originalData_df.groupby('TYPE').apply(sampling_k_elements).reset_index(drop=True)
originalData_df.head()
```

Furthermore, for the second dataset I found that each disease presented with the same 8-10 symptoms every time. This makes dimensionality reduction simple as all I had to do was find any symptom that presents itself in at least on of the four diseases, but not in all four. This works because features are binary and all symptoms present nearly

uniformly for a disease which means there is little to no information gain from using a symptom that displays in all four diseases.

```
def isUnique(col, df1, df2, df3, df4):
    if df1.loc[col] > 0 and df2.loc[col] > 0 and df3.loc[col] > 0 and df4.loc[col] > 0:
        return False
    return True
```

```
columns = set()

for col in flu_df.items():
    if isUnique(col[0], flu_df, allergy_df, cold_df, covid_df):
        columns.add(col[0])

for col in allergy_df.items():
    if isUnique(col[0], flu_df, allergy_df, cold_df, covid_df):
        columns.add(col[0])

for col in cold_df.items():
    if isUnique(col[0], flu_df, allergy_df, cold_df, covid_df):
        columns.add(col[0])

for col in covid_df.items():
    if isUnique(col[0], flu_df, allergy_df, cold_df, covid_df):
        columns.add(col[0])
print(columns)
```

```
{'NAUSEA', 'STUFFY_NOSE', 'DIARRHEA', 'LOSS_OF_SMELL', 'LOSS_OF_TASTE', 'ITCHY_NOSE', 'VOMITING', 'SHORTNESS_OF_BRE
ATH', 'ITCHY_MOUTH', 'DIFFICULTY_BREATHING', 'ITCHY_INNER_EAR', 'RUNNY_NOSE', 'PINK_EYE', 'ITCHY_EYES', 'FEVER'}
```

Implementation

For the vitals dataset, our data was split into a train and test dataset, with 25% of all examples withheld for testing. A simple Logistic Regression and Decision Tree from Scikit-Learn were used for training. No further parameters were passed into these models. Next I trained both models using the same training set, but without the temperature feature. Again, no parameters were passed into the algorithms, but for these two models I used the Scikit-learn Pipeline functionality to combine both the standard feature scaler and the model. This is important if I want to implement either of these models on an iOS app using CoreML. We will need an easy way to scale readings from an Apple Watch similar to the scaling we have while training.

```
scaler = preprocessing.StandardScaler()
lm2 = LogisticRegression(random_state=0)

pipeline = Pipeline([('scaler', scaler), ('logistic_model2',lm2)])

logistic_model2 = pipeline.fit(X_train2, y_train2)
```

For the second dataset, I first split the dataset again between a test and train dataset. I only withheld 15% of the records for testing this time, as I figured we can test later on with the entirety fo the dataset not originally changed in order to balance it. I first

attempted to use a Random Forest model with 5 estimators and an entropy criterion. I also built an SVM model with an 'rbf' kernel and a C = 1.0.

Refinement

Little refinement was done throughout this process. Among some of the changes I did end up making to improve my models, for my first vitals model, I decided a good refinement was to take the confidence interval outputs from the Logistic Regression and only classify results as negative if probability of being negative was higher than 75%. This created a high sensativity with minimum change to the specificity.

```
y_pred_raw = [logistic_model2.predict_proba([x]) for x in X_test2]
y_pred = [0 if x[0][0] > 0.75 else 1 for x in y_pred_raw]
```

Second, for my other dataset, I found that while both the random forest and SVM were good at learning the balanced dataset (both training and test subsets), they were not able to generalize to the entire unbalanced dataset. Sensitivity and specificity were nearly perfect for allergies and the Flu, but precision was only about 50% for both Covid-19 and the cold. In order to resolve this, the first thing I did was change the number of example classification from 2048 to 5000. I originally choose 2048 because this would encompass all of the Covid-19 cases, but it appears this did not allow for enough cases of Flu and Allergy to fully be able to differentiate. I decided to almost double that number to 5000, although the total number of cases for the Cold and Covid-19 would remain the same as they only had 1024 and 2048 cases respectively. This increase had a dramatic improvement on the final results.

IV. Results

Model Evaluation and Validation

Our first vitals decision tree model that looked at all three features ended up having an accuracy score of 100% on both the train and test dataset. The logistic model I tried had an accuracy score of 92% for both the training and test dataset. While this is lower than the decision tree, it is in the sweet spot of being high without initially indicating an overfitting of the dataset. While these results are really promising, I was more interested in the results of our models that do not look at temperature. Both models ended up having an accuracy of 85% on both the train and test dataset. The decision tree model more specifically had a Specificity of 87%, but sensitivity was only 84%

```
Specificity: 0.8746987951807229, Sensitivity: 0.8438247011952191
```

But, for my logistic regression model, If I change the threshold of what is a positive case to only those where the model is higher than 68% certain the user does not have Covid-19, then my sensitivity is as high as 91%. Although I do have a tradeoff in that my specificity is 7% lower for the logistic regression model over the decision tree, this is a welcome tradeoff for the task at hand. False negatives are much more serious than a false positive as a false negative lulls users into potentially not seeking treatment and acting in ways that may infect others.

```
Specificity: 0.8, Sensitivity: 0.9147410358565737
```

For the second dataset, I choose to use a confusion matrix to help me evaluate the models. For the Random Forest, my confusion matrix shows perfect precision for all four categories. Most importantly, recall for all diseases was very high.

```
{'ALLERGY': 0, 'COLD': 1, 'COVID': 2, 'FLU': 3}
[[137   0   0   0]
 [  0 156   0   0]
 [  0   0 155   0]
 [  0   0   0 167]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       137
           1       1.00      1.00      1.00       156
           2       1.00      1.00      1.00       155
           3       1.00      1.00      1.00       167

 avg / total       1.00      1.00      1.00       615
```

Next I take this model and test it against the entire unbalanced dataset. I find even in this case that the results are perfect.

```
{'ALLERGY': 0, 'COLD': 1, 'COVID': 2, 'FLU': 3}
[[16381     0     0     0]
 [    0  1024     0     0]
 [    0     0  2048     0]
 [    0     0     0 25000]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     16381
           1       1.00      1.00      1.00      1024
           2       1.00      1.00      1.00      2048
           3       1.00      1.00      1.00     25000

 avg / total       1.00      1.00      1.00     44453
```

Next, for the SVM I found similarly high precision and recall for all categories.

```
{'ALLERGY': 0, 'COLD': 1, 'COVID': 2, 'FLU': 3}
[[137   0   0   0]
 [  0 156   0   0]
 [  0   0 155   0]
 [  0   0   0 167]]
             precision    recall  f1-score   support

          0       1.00      1.00      1.00       137
          1       1.00      1.00      1.00       156
          2       1.00      1.00      1.00       155
          3       1.00      1.00      1.00       167

avg / total       1.00      1.00      1.00       615
```

But once again, when we run our model against the entire unbalanced dataset, we find that the data does not seem complicated enough to be a problem for our model to perfect.

```
{'ALLERGY': 0, 'COLD': 1, 'COVID': 2, 'FLU': 3}
[[16381     0     0     0]
 [    0  1024     0     0]
 [    0     0  2048     0]
 [    0     0     0 25000]]
             precision    recall  f1-score   support

          0       1.00      1.00      1.00     16381
          1       1.00      1.00      1.00      1024
          2       1.00      1.00      1.00      2048
          3       1.00      1.00      1.00     25000

avg / total       1.00      1.00      1.00     44453
```

Justification

As I have mentioned, this project requires the use of two distinct models in order to provide insight into possible Covid-19 infections. The first model is supposed to

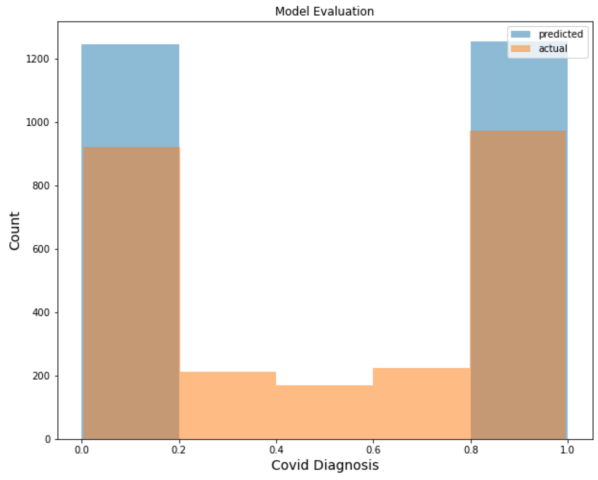
constantly monitor user vitals without much needed from them. The second model is meant to ask users a bunch of questions to further differentiate between different infections. Part of the justification for this is that a tradeoff for the ease of use for the first model is more inaccuracy than the second model. Our findings show this to be true. When moving this into production, I would look to utilize the Logistic Regression model that uses only Sp02 and Pulse data as the first model for our project. Although the specificity was only 80%, the sensitivity is very high. Considering that our benchmark is a pcr test which similarly tends to produce many false positives but few false negatives, our model seems to be sufficient for our need. Also, sine we have a second model that can further evaluate user symptoms, a false positive only requires a few minutes of answering questions to get further insight into their health. The 80% specificity we reached is actually slightly higher than the 72% PCR result mentioned above for those displaying symptoms. More research would be needed to know what it means to not display symptoms and if that impacts Sp02 and pulse readings. It might be that non-symptomatic individuals do not experience changes in Sp02 and pulse and therefore our model will almost always fail in those scenarios.

Our second model was able to effectively differentiate between cold, Flu, allergies, and Covid-19. The perfect precision and recall are actually a cause for concern, as they hint of overfitting to the dataset. Further data and research would be needed to know if this is in fact an issue or if the problem at hand was simple and compartmentalized enough for our model to be so accurate.

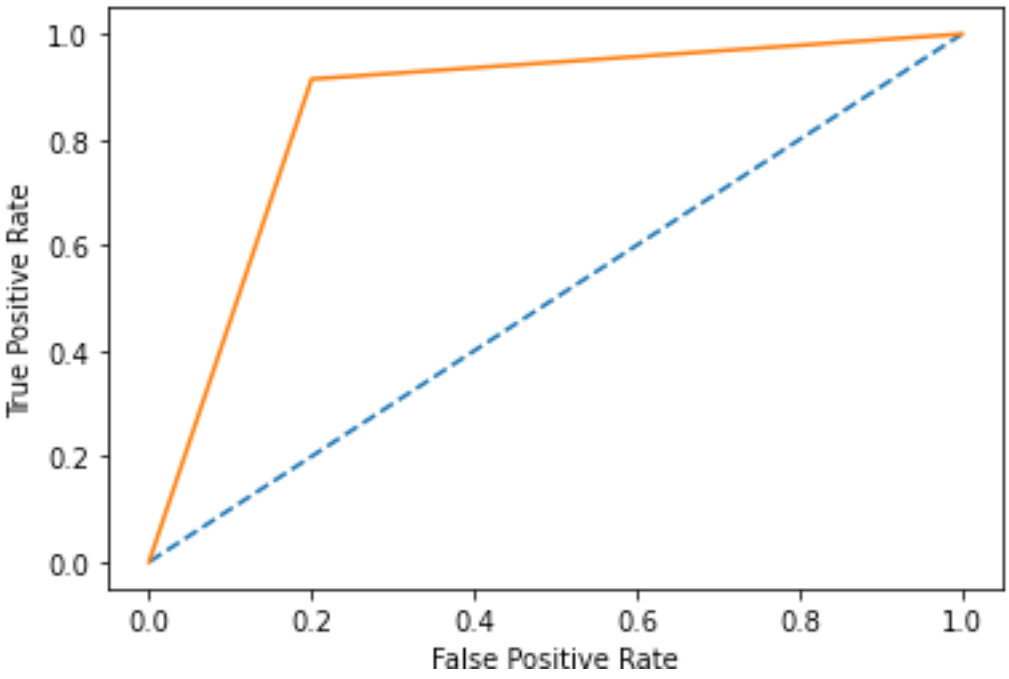
V. Conclusion

Free-Form Visualization

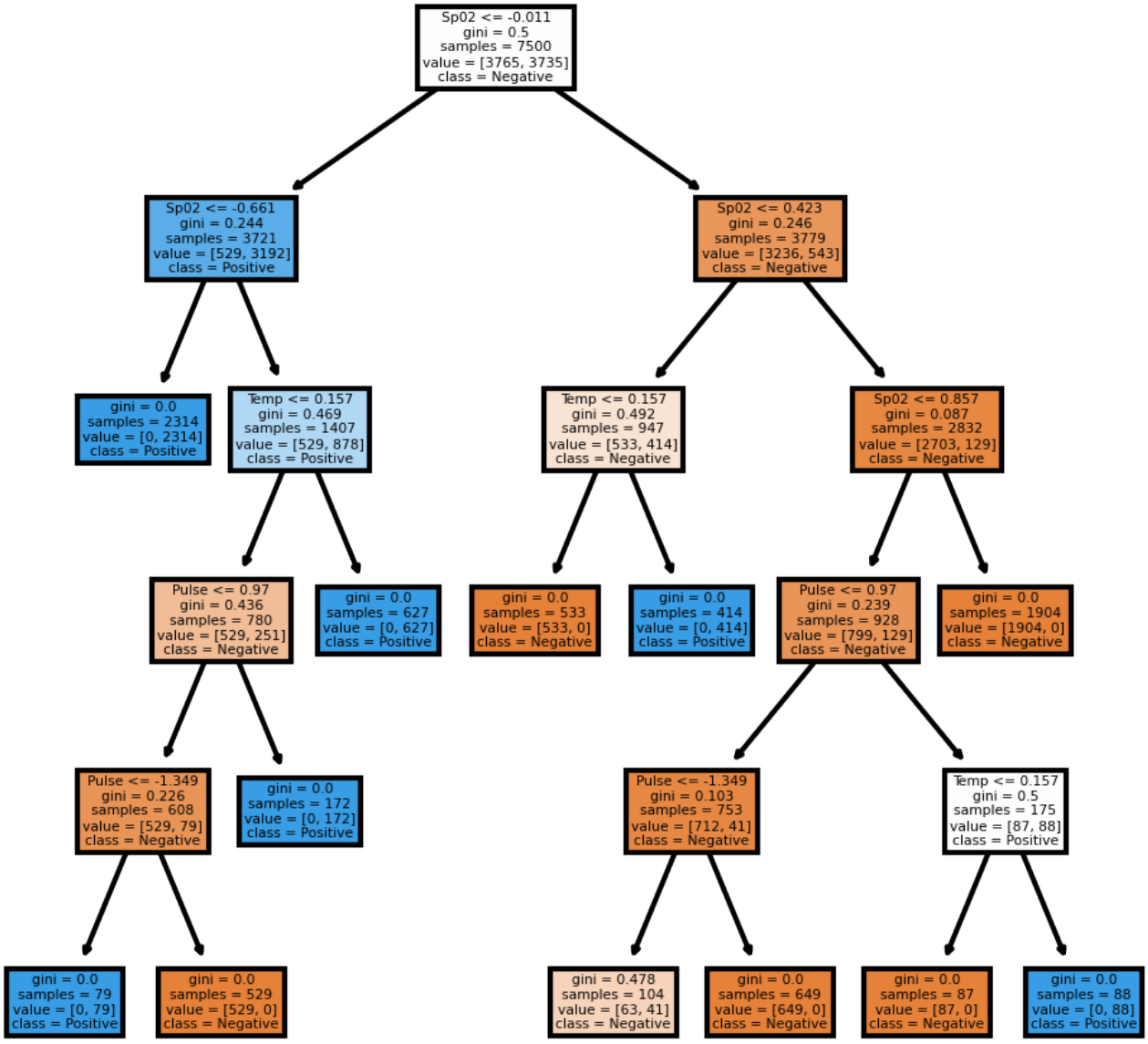
Our first model will be utilized to monitor user health throughout the day using an iOS app that runs in the background. This logistic regression model will be outputting negative or positive classifications with confidence percentages. Based on my evaluation of the model, I found that setting a negative outcome threshold at 68% produced the best sensitivity relative to the specificity. If we plot the models output over the entire test set and compare it to the actual values, we see that the model tends to be very confident in its predictions. Most examples are predicted below 0.2 or above 0.8.



I also plotted an roc curve as another way of evaluating our logistic regression model. An roc curve helps us visualize the performance of our classification. This helps us understand how well the model can distinguish between classes. For our chart, the higher that orange line is (the AUC line), the better the model is at predicting negative examples as negative and positive examples as positive. As can be seen, our model has a pretty high auc and indicates our model is pretty performant.



Although I choose to go with the logistic regression model for the vitals dataset over the decision tree. One of the things I like about decision trees are that it is easy to see how they come to their decisions. We can plot out the tree and gather insights for ourselves into how Covid-19 symptoms present themselves and how we can better recognize and diagnose them ourselves. Below is the decision tree model used for Sp02, pulse, and temperature.



I also did go ahead and build out a demo iOS app that uses our first logistic regression model to monitor Sp02 and pulse data from an Apple Watch to monitor for Covid-19[5][6].

Reflection

Overall, I found this project to be a great success. The goal is to begin moving health monitoring onto devices that are becoming more prevalent in modern society. As more sensors get added and these sensors become more accurate, the need to build

---

[5] The repo can be found here: https://github.com/crweaver225/Covid-Monitor

[6] A video demo of the app can be found here: https://youtu.be/FoL-Q-pcomY

models that can catch issues and better inform users of their own health will become more necessary. Wearables will probably never be as accurate as more invasive tests, but considering the low cost implementation of monitoring, this should not be our benchmark. One of my major concerns was would our models be able to successfully help monitor for Covid-19 without any user interaction with the models. This is not possible at the moment if temperature is a required feature, as no sensors are currently available as an api to developers to measure temperature. But what I found is our results are still acceptable when only measuring Sp02 and pulse. Realistically, pulse is not all that helpful of a feature and provides little insight to our model. But because our dimensions are already so low, any further information we can feed the model to slightly improve it will be used.

Our second model surprised me as far as how accurate it was. I expected different diseases would present themselves differently as far as symptoms go, but apparently the demarcation is clear enough for our models to almost always find the correct answer. I was surprised to see that the same symptoms tend to present themselves almost uniformly for each disease. For example, for Covid-19, all 11 symptoms that are presented show between 491 and 525 times. The deviation between these is small and hints to me that I either don't have a great grasp on how these diseases work or that something may be off with the dataset. But given what we have as our data, the model was very accurate and successfully does its job.

The importance for our second model is that the first model may not be able to differentiate between Flu, Cold, and Covid-19 if they all impact Sp02 and pulse. Because the first model is so low cost to use and produces a decent amount of false positives, this second model is a quick way to further clarify what is going on with the end user.

Improvement

The biggest areas of improvement revolve around the datasets we used and how they apply to the real world. My first concern is that I do not know how Covid-19 positive cases were classified in the datasets. If these cases were confirmed via PCR tests then the same inaccuracies that plague those tests will also plague my models. Further, PCR tests tend to be far more inaccurate when users are not displaying any symptoms. What it means to display symptoms and how those get translated into Sp02 and pulse biometrics is not clear. Further research is needed as it may be the case that non-symptomatic individuals with Covid-19 do not demonstrate Sp02 and pulse readings that indicate a positive case. That would mean our models will be extremely inaccurate in those user cases.

I also found my second dataset to be suspect in its distribution of symptoms for each disease. There is very little variability for the symptoms that present per disease, which is not what I would expect. For example, I would guess nausea presents in about half or less of Covid-19 cases whereas difficultly breathing would display much more frequently. What I found was these presented about the same in aggregate. Further

research into if this is an issue with the dataset is important to understand how well our models will work in the real world.

Finally, one major disadvantage of our second model is that it does not handle, as a classification, that the user does not have either a cold, Flu, allergies, or Covid-19. Because there are other diseases and considerations in the real world, we ideally should have another category for ‘Other’ that acts as a catch all for use cases where the model does not have a high confidence of categorizing elsewhere. For example, It might be that a user is at an altitude higher than they normally experience which might influence their Sp02 and pulse readings into producing a false positive for our first model. In this case, our second model should be able to look through the lack of symptoms and inform the user that a diagnosis is none of the above for cold, Covid-19, flu, or allergies.