Project Report

-Ryan L'Abbate
-Xingrou Mei
-Antonio Singh
-Yuwen Yu

Project Goals

There were two main goals of this project:

1. Develop models to accurately classify U.S. counties based on COVID positivity and fatality rates
2. Identify predictors of COVID positivity and fatality rates in the United States

Preprocessing

Raw Datasets:

Two raw datasets were used for this project.

1. 2016 US Presidential Election Data: 3149 rows x 159 columns
2. COVID Data (2021): 384,930 rows x 7 columns

The 2016 US Presidential Election data contains demographic information about various counties of the US such as education level, income, occupation, and race as well as political information.

The second is the COVID dataset obtained from opendatasoft (https://public.opendatasoft.com/explore/dataset/coronavirus-covid-19-pandemic-usa-counties/export/?disjunctive.province_state&disjunctive.admin2&refine.date=2021%2F04). This dataset contains COVID statistics about counties across the US such as number of cases, number of deaths, COVID positivity rate, and COVID fatality rate as of 4/27/21. The dataset also includes latitude and longitude for each county.

Missing Values

The raw datasets were examined for missing values in the columns of interest. Political information from the election data was not included due to the large increase in dimensionality it would cause.

Election Data Missing Values			
Field	Percent Missing Values	Field	Percent Missing Values
State	0	Construction.extraction.maintenance.and.repair.occupations	0
Fips	0	Production.transportation.and.material.moving.occupations	0
County	0	White	0
Precincts	1.018136	Black	0
Less Than High School Diploma	0	Hispanic	0
At Least High School Diploma	0	Asian	0
At Least Bachelor's Degree	0	Amerindian	0
Graduate Degree	0	Other	0
School Enrollment	0	White Asian	0
Median Earnings 2010	0	Median Age	0
Children Under 6 Living in Poverty	0	Poor.physical.health.days	10.785873
Adults 65 and Older Living in Poverty	0	Poor.mental.health.days	17.721922
Total Population	0	Adult.smoking	13.84028
Poverty.Rate.below.federal.poverty.threshold	0	Adult.obesity	0.159084
Gini.Coefficient	0	Diabetes	0.159084
Management.professional.and.related.occupations	0	Sexually.transmitted.infections	6.013363
Service.occupations	0	HIV.prevalence.rate	26.12154
Sales.and.office.occupations	0	Uninsured	0.1909
Farming.fishing.and.forestry.occupations	0	Unemployment	0.1909

HIV.prevalence.rate was dropped from the dataset due to its high percentage of missing values (26.1%) and the presence of a similar column (Sexually.transmitted.infections). All other missing values were filled with the column means.

The Covid dataset had no missing values in the fields chosen for this project analysis.

```
Covid Data Missing Value Percents:
Admin 2 FIPS Code    0.0
Date                 0.0
Total Death          0.0
Total Confirmed      0.0
Fatality Rate        0.0
location             0.0
dtype: float64
```

Most Recent Entry for each County in COVID Dataset

The covid dataset was filtered so that only the most recent entry for each FIPS code was included. The election dataset only gives us county demographic data at one point in time so joining the same data to the same FIPS code at multiple points in time is redundant. This reduced the COVID dataset to 3,280 rows, roughly equal to the election dataset.

Split Location Field to Latitude and Longitude

The location field was split from one column containing a tuple into two columns for latitude and longitude. Latitude and longitude will be used to plot the data on a map.

Joining the Datasets

The election dataset and COVID dataset were joined together using an inner join on the FIPS code, a field found in both datasets.

Positivity Rate & Fatality Rate

A column for Positivity Rate was added to the dataset. This value will be used to define classes for the data. The column was calculated as follows:

Positivity Rate = Confirmed Cases / Total Population

A field for fatality rate was already provided in the raw COVID dataset.

The national average for positivity rate is 9.93%.
The national average for fatality rate is 1.95%.

2-Class Definition

The fields ‘high risk positivity ’ and ‘high risk fatality’ were created as a 2-class categorization for the data.

Class 0: positivity/fatality rate less than or equal to the national average
Class 1: positivity/fatality rate greater than the national average

Positivity Rate Class Counts		
Class	**Positivity Rate Count**	**Value Range (Positivity)**
0	1567	<= 0.0993
1	1573	> 0.0993

Fatality Rate Class Counts		
Class	**Fatality Rate Count**	**Value Range (Fatality)**
0	1779	<= 0.0195
1	1361	> 0.0195

6-Class Definition

The fields ‘risk level positivity’ and ‘risk level fatality’ were created as a 6-class categorization for the data.

The six classes partition data by positivity/fatality rate based on thresholds of the average plus or minus 1 or 2 standard deviations.

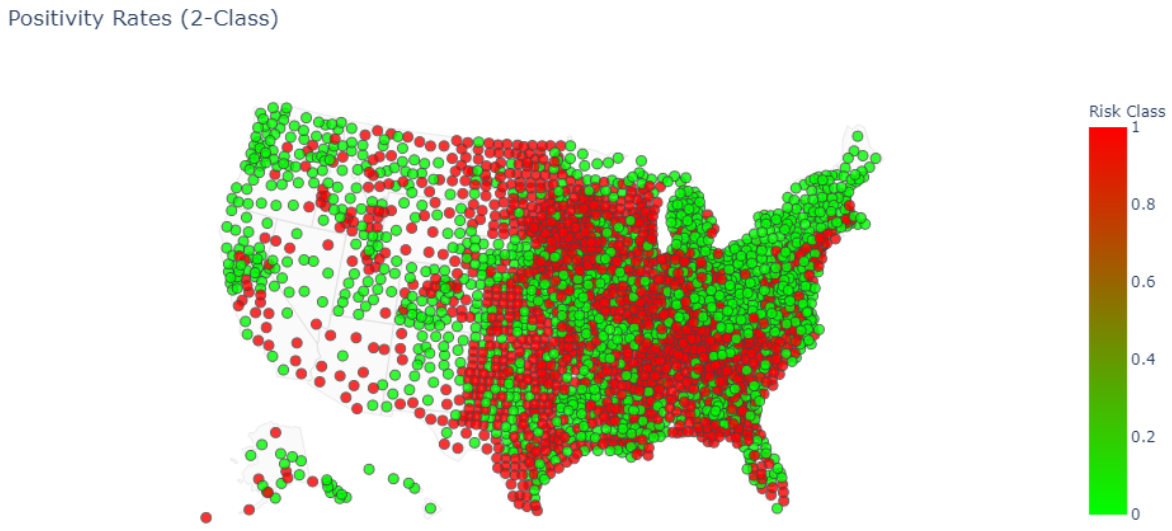
Positivity Rate Class Counts			
Class	**Count**	**Class Definition**	**Values**
0	93	< AVG - 2*STD	< 0.0350
1	309	Between AVG - 2*STD and AVG - STD	0.0350 to 0.0672
2	1165	Between AVG - STD and AVG	0.0672 to 0.09934
3	1194	Between AVG and AVG + STD	0.0994 to 0.1315
4	294	Between AVG + STD and AVG + 2*STD	0.1315 to 0.16371
5	85	> AVG + 2*STD	> 0.1637

Fatality Rate Class Counts			
Class	**Count**	**Class Definition**	**Values**
0	0	< AVG - 2*STD	< -0.0009
1	354	Between AVG - 2*STD and AVG - STD	-0.0009 to 0.0093
2	1425	Between AVG - STD and AVG	0.0093 to 0.0196
3	935	Between AVG and AVG + STD	0.0196 to 0.0298
4	317	Between AVG + STD and AVG + 2*STD	0.0298 to 0.0401
5	109	> AVG + 2*STD	> 0.0401

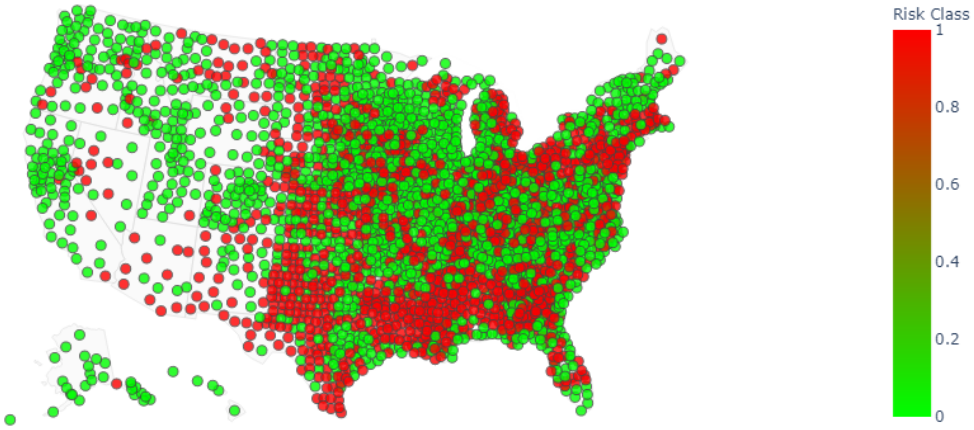
The fatality rate is noticeably less normally-distributed than the positivity rate, with the lowest class threshold falling below 0 and the class counts being less symmetric around the average.

Plotting the Data on Maps

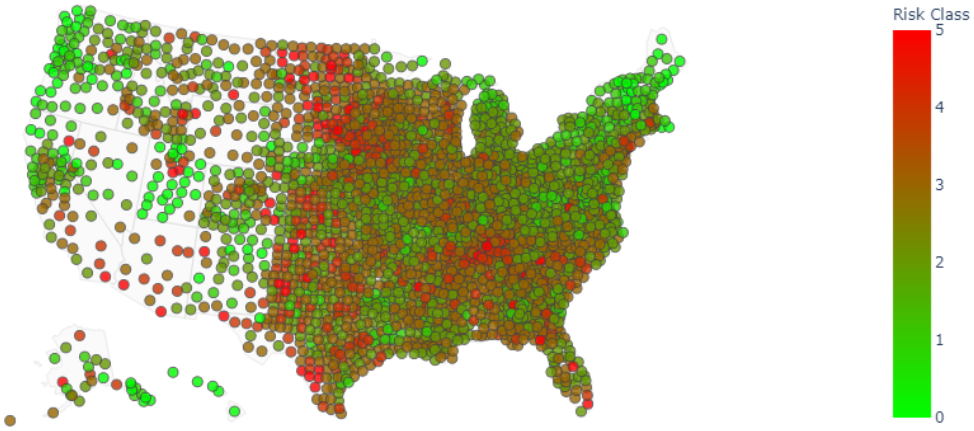
The latitude and longitude fields were used to plot the data onto a map of the US. The data was color-coded with red indicating high positivity rate/risk and green indicating low positivity rate/risk. Three plots were made for positivity and fatality rate, one for 2-class data, one for 6-class data,and one using the positivity rate itself instead of any classes.



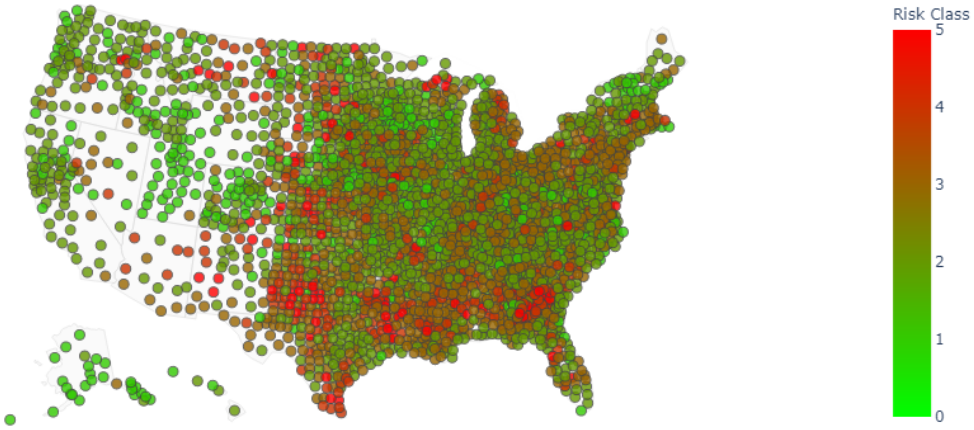
Fatality Rates (2-Class)



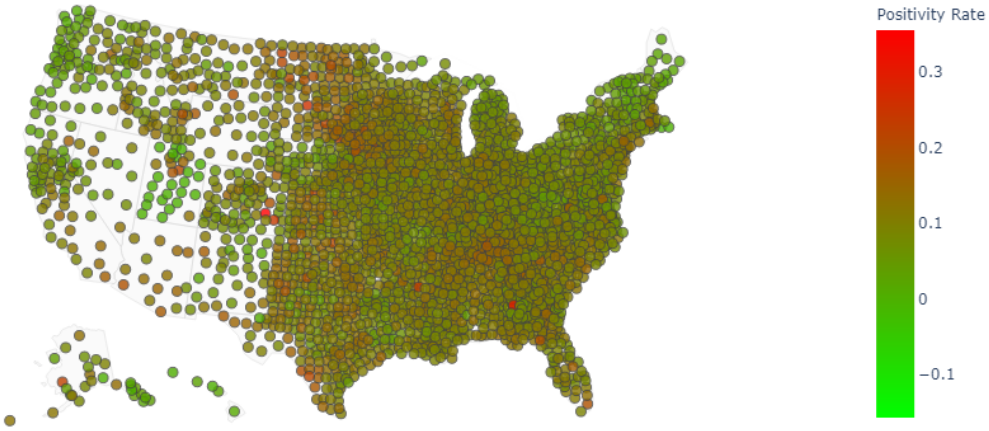
Positivity Rates (6-Class)

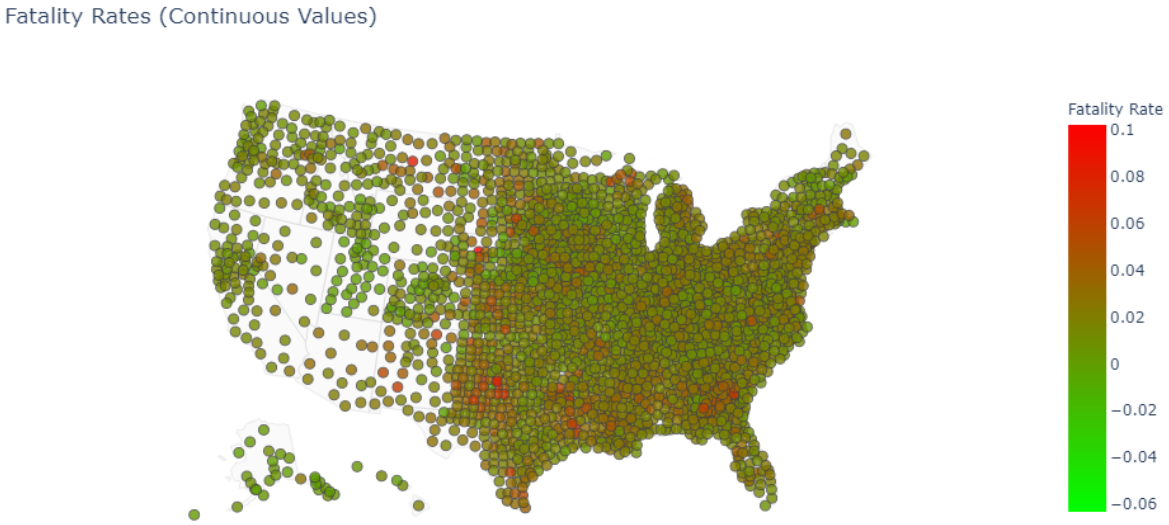


Fatality Rates (6-Class)

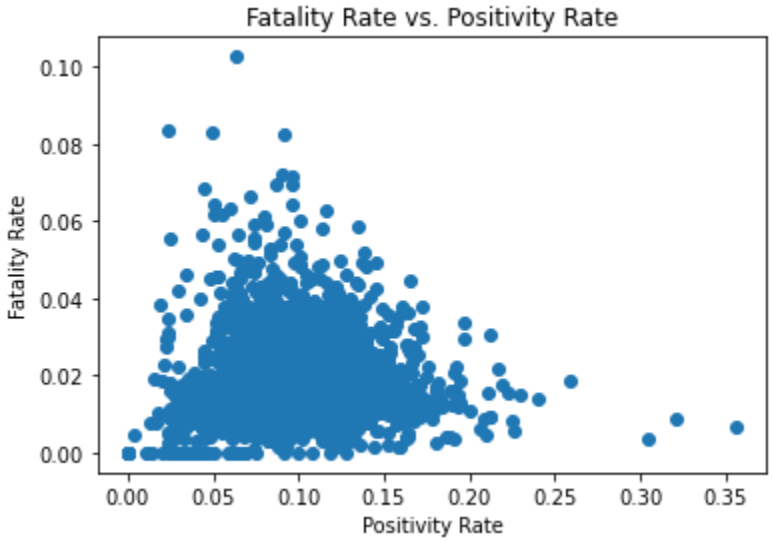


Positivity Rates (Continuous Values)





Graph of Fatality Rate vs. Positivity Rate



This graph suggests that positivity rate and fatality rate are not strongly correlated. This could indicate that the two values have different causes.

Normalization

As the final preprocessing step, all continuous variables were normalized to range of 0 to 1. This will help the models process the data more efficiently and make it easier to compare variable

correlations and determine which variables are the best predictors of positivity rate and fatality rate.

The final dataset is 3,140 rows x 46 columns.

Models:

- **Logistic Regression**

Before comparing which models will perform better for this COVID-19 dataset, there should be a baseline to help us understand the dataset better. With a baseline, we can discover which classes are harder to separate, and which features are more important. It also sets a guideline to improve the model based on the observations from a basic model. The baseline for this project is logistic regression. There are four models for logistic regression, two multiclass and two binary classifications using the positivity rate and the fatality rate. Before training the model, we drop the state, county, fips, date, and positivity rate when predicting the positivity class and fatality rate when predicting the fatality class. To train the models, first, the data split the testing data and training data to 0.1 and 0.9 accordingly. Then, set a random state to get the same test set every run.

Binary Classification: Positivity Rate

The testing accuracy for the 2-class classification with the positivity rate is 70.38%, and the training accuracy is 70.10%.

Confusion Matrix:

		Predicted	
		0	1
Actual	0	111	40
	1	53	110

According to the confusion matrix, the number of correctly predicted instances vs. wrong instances is higher for both classes 0 and 1, and it is a good sign. When predicting which county is high risk and which county is low risk, the false-negative part is essential. It indicates that places that are supposed to be at high risk but predicted as low risk. It can cause a problem when it comes to COVID infection.

Classification Report:

The recall score for class 0 is 74% and for class 1 is 67%. Overall, this model performs not that good because it is only 67% of high-risk correctness.

Classification Report

	precision	recall	f1-score	support
0.0	0.68	0.74	0.70	151
1.0	0.73	0.67	0.70	163
accuracy			0.70	314
macro avg	0.71	0.70	0.70	314
weighted avg	0.71	0.70	0.70	314

Weight on features:

As shown below, the feature with the most weight on positivity rate is the Gini Coefficient, which is the distribution of income by county in this case. Other than the Gini Coefficient, features with a high impact on the positivity rate are people with less than a high school diploma, occupations in production, transportation, and material moving, Amerindian, Total confirmed, etc. It makes sense why production and transportation have a high impact on the positivity rate since they are essential workers and constantly moving from place to place. They have a higher chance of catching a covid. Features with a negative weight on the positivity rate are graduate degree, median age, unemployment, etc. People staying home more often may be the reason why unemployment has less impact on the positivity rate.

Below is an entire list of weight for all features:

Gini.Coefficient	2.692240
Less Than High School Diploma	2.348532
Production.transportation.and.material.moving.occupations	2.234238
Amerindian	2.233836
Total Confirmed	1.533001
Adult.obesity	1.480317
Adults 65 and Older Living in Poverty	1.374347
Sexually.transmitted.infections	1.365952
Total Death	1.196754
Sales.and.office.occupations	0.963653
At Least Bachelors's Degree	0.858170
Hispanic	0.670128
Median Earnings 2010	0.650514
School Enrollment	0.489561
White	0.148674

Total Population	-0.204548
Uninsured	-0.234097
Longitude	-0.286271
Precincts	-0.304424
Service.occupations	-0.381693
Children Under 6 Living in Poverty	-0.550054
White Asian	-0.553414
Poor.physical.health.days	-0.690504
Diabetes	-0.760189
Management.professional.and.related.occupations	-0.795809
At Least High School Diploma	-0.862480
Other	-0.982011
Poor.mental.health.days	-1.176672
Fatality Rate	-1.188324
Asian	-1.588666
Black	-1.702991
Poverty.Rate.below.federal.poverty.threshold	-1.728566
Adult.smoking	-1.735058
Latitude	-2.031665
Farming.fishing.and.forestry.occupations	-2.105335
Construction.extraction.maintenance.and.repair.occupations	-2.106922
Unemployment	-2.819728
Median Age	-3.180905
Graduate Degree	-4.095708

Binary classification: Fatality Rate

The testing accuracy for the 2-class classification with the fatality rate is 71.34%, and the training accuracy is 68.65%.

Confusion Matrix:

		Predicted	
		0	1
Actual	0	149	59
	1	31	75

According to the confusion matrix, the number of correctly predicted for class 0 is higher than class 1. For class 1, there are only 75 predicted correctly.

Classification Report:

The recall score for class 0 is 72% and for class 1 is 71%. Overall, the model using the fatality rate performs better than using the positivity rate.

Classification Report

	precision	recall	f1-score	support
0.0	0.83	0.72	0.77	208
1.0	0.56	0.71	0.62	106
accuracy			0.71	314
macro avg	0.69	0.71	0.70	314
weighted avg	0.74	0.71	0.72	314

Weight on features:
As shown below, the feature with the most weight on the fatality rate is the median age. Other than the median age, features with a high impact on the fatality rate are diabetes, school enrollment, longitude, Gini Coefficient, total death, etc. Based on the weight, the Gini Coefficient has a high impact on both positivity rate and fatality rate classifications. It is interesting to see how school enrollment has a high impact on the fatality rate. Features with a negative weight on the fatality rate are at least a bachelor's degree, latitude, occupations in construction, extraction, maintenance, repair, sales, and office.

Median Age	3.830289
Diabetes	2.474467
School Enrollment	2.024323
Longitude	1.875049
Gini.Coefficient	1.528629
Total Death	1.429731
Hispanic	1.311868
Children Under 6 Living in Poverty	0.836583
Management.professional.and.related.occupations	0.675038
Adults 65 and Older Living in Poverty	0.644732
Sexually.transmitted.infections	0.589235
Production.transportation.and.material.moving.occupations	0.483886
Uninsured	0.453224
Median Earnings 2010	0.403614
Amerindian	0.334617
Less Than High School Diploma	0.224782
Service.occupations	0.213698
Adult.obesity	0.195941
Black	0.154577
At Least High School Diploma	-0.044569
Poor.mental.health.days	-0.072366
Precincts	-0.183310
Asian	-0.244215
Adult.smoking	-0.306481
Unemployment	-0.399855
Farming.fishing.and.forestry.occupations	-0.405445
White	-0.412760
Poverty.Rate.below.federal.poverty.threshold	-0.502441
White Asian	-0.521139
Total Population	-0.581396
Graduate Degree	-0.595670
Poor.physical.health.days	-0.667575
Positivity Rate	-0.889052
Total Confirmed	-0.891019
Sales.and.office.occupations	-0.977704
Construction.extraction.maintenance.and.repair.occupations	-1.240038
Latitude	-1.752900
Other	-2.080273
At Least Bachelors's Degree	-2.602708

Multiclass: Positivity Rate:

For multiclass, the test set is changed to 0.20 due to a better result with more testing data. The testing accuracy for the 6-class classification with the positivity rate is 49.52%, and the training accuracy is 51.51%.

Confusion Matrix:

		Predicted					
		0	1	2	3	4	5
Actual	0	1	1	2	0	0	0
	1	4	6	6	1	0	0
	2	8	43	142	64	12	4
	3	5	9	92	162	40	20
	4	0	1	0	3	0	1
	5	0	0	0	0	1	0

According to the multiclass confusion matrix, there are not many correctly predicted for all classes. Class 2 and Class 3 predicted the most and with the most correctly predicted number, which is in the mid level of risk. There are not many false-negatives with a significant difference.

Classification Report:
Based on the classification report, the recall score for all classes is lower than 50%, except class 2 with 2% more. Overall, logistic regression does not perform well in multiclass.

Classification Report

	precision	recall	f1-score	support
0	0.06	0.25	0.09	4
1	0.10	0.35	0.16	17
2	0.59	0.52	0.55	273
3	0.70	0.49	0.58	328
4	0.00	0.00	0.00	5
5	0.00	0.00	0.00	1
accuracy			0.50	628
macro avg	0.24	0.27	0.23	628
weighted avg	0.63	0.50	0.55	628

Weight on features:

As shown below, the common features with a high impact on the positivity rate for lower risk classes are median age. Higher risk classes are less than high school diplomas. However, the less impact for lower risk classes is less than high school diplomas. For higher-risk classes, it is the

median age. They are the opposite of each other. There are other features that are the opposite of each other based on the higher or lower class. Such as occupations in construction, extraction, maintenance, and repair, Gini coefficient, etc.

Below is a full list for Class 0:

Construction.extraction.maintenance.and.repair.occupations	1.720069
Other	1.604788
Median Age	1.457977
At Least High School Diploma	1.408414
Graduate Degree	1.137930
Farming.fishing.and.forestry.occupations	1.125672
Poor.mental.health.days	1.052332
Unemployment	1.032456
Poor.physical.health.days	0.897603
Asian	0.694788
Service.occupations	0.569057
White Asian	0.464035
Amerindian	0.420481
Latitude	0.389927
Diabetes	0.232740
White	0.156752
At Least Bachelors's Degree	0.089146
Management.professional.and.related.occupations	-0.213942
School Enrollment	-0.227993
Total Death	-0.250938
Total Population	-0.286343
Precincts	-0.324212
Poverty.Rate.below.federal.poverty.threshold	-0.326455
Total Confirmed	-0.344460
Sales.and.office.occupations	-0.346156
Uninsured	-0.404830
Sexually.transmitted.infections	-0.506290
Adult.smoking	-0.667145
Adults 65 and Older Living in Poverty	-0.772671
Hispanic	-0.832283
Median Earnings 2010	-0.938362
Gini.Coefficient	-0.939063
Children Under 6 Living in Poverty	-1.071860
Black	-1.076154
Production.transportation.and.material.moving.occupations	-1.197435
Adult.obesity	-1.400988
Fatality Rate	-2.079834
Less Than High School Diploma	-2.245427
Longitude	-2.745822

Below is a full list for Class 1:

Median Age	2.763505
Unemployment	2.721739
Latitude	2.708373
Graduate Degree	1.969666
Farming.fishing.and.forestry.occupations	1.835430
Adult.smoking	1.333629
Construction.extraction.maintenance.and.repair.occupations	1.187605
Fatality Rate	1.119048
Poor.physical.health.days	1.119006
Black	1.118296
Children Under 6 Living in Poverty	0.872752
Asian	0.848027
At Least Bachelors's Degree	0.657159
Longitude	0.584136
At Least High School Diploma	0.535672
Management.professional.and.related.occupations	0.497354
Median Earnings 2010	0.466508
Diabetes	0.426213
Poverty.Rate.below.federal.poverty.threshold	0.295211
Poor.mental.health.days	0.286736
Other	0.217875
Service.occupations	0.206135
White Asian	0.025065
Uninsured	-0.010504
Less Than High School Diploma	-0.201360
Hispanic	-0.226564
Total Population	-0.254949
White	-0.349436
Sales.and.office.occupations	-0.408233
Precincts	-0.435289
Adults 65 and Older Living in Poverty	-0.643612
School Enrollment	-0.708910
Total Confirmed	-0.811160
Total Death	-0.916385
Gini.Coefficient	-0.941456
Amerindian	-1.045526
Sexually.transmitted.infections	-1.239357
Adult.obesity	-1.291969
Production.transportation.and.material.moving.occupations	-1.631833

Below is a full list for Class 2:

Longitude	1.833478
Fatality Rate	1.744619
Black	1.581093
Graduate Degree	1.332433
Poverty.Rate.below.federal.poverty.threshold	1.283339
Adult.smoking	1.216396
Poor.mental.health.days	0.884446
Precincts	0.838283
Median Age	0.739384
School Enrollment	0.628552
Diabetes	0.628412
Total Population	0.615668
Children Under 6 Living in Poverty	0.538324
Management.professional.and.related.occupations	0.466241
White Asian	0.461611
Asian	0.419753
Uninsured	0.344787
Construction.extraction.maintenance.and.repair.occupations	0.299223
Unemployment	0.297465
White	0.275770
Adult.obesity	0.246429
At Least High School Diploma	0.189471
Farming.fishing.and.forestry.occupations	0.099716
Sexually.transmitted.infections	0.099536
Hispanic	0.078272
Production.transportation.and.material.moving.occupations	-0.052701
Median Earnings 2010	-0.121957
Total Death	-0.173649
Service.occupations	-0.290719
Total Confirmed	-0.429064
Poor.physical.health.days	-0.506747
Latitude	-0.514143
Sales.and.office.occupations	-0.754510
Other	-0.789819
Adults 65 and Older Living in Poverty	-0.950605
At Least Bachelors's Degree	-0.995054
Less Than High School Diploma	-1.238608
Gini.Coefficient	-1.336807
Amerindian	-1.676133

Below is a full list for Class 3:

Production.transportation.and.material.moving.occupations	2.253307
School Enrollment	1.682028
Sexually.transmitted.infections	1.458596
Gini.Coefficient	0.920730
Total Death	0.879507
Adult.obesity	0.867827
Median Earnings 2010	0.684545
Total Confirmed	0.607119
Less Than High School Diploma	0.595431
Longitude	0.370615
Black	0.339431
At Least Bachelors's Degree	0.316301
Poor.mental.health.days	0.307737
Diabetes	0.239437
Precincts	0.194942
Adults 65 and Older Living in Poverty	0.187358
Fatality Rate	0.162071
Amerindian	0.138205
Hispanic	0.117330
Total Population	0.037238
White	-0.021061
Uninsured	-0.106420
At Least High School Diploma	-0.115781
Adult.smoking	-0.212711
Children Under 6 Living in Poverty	-0.243655
White Asian	-0.306079
Sales.and.office.occupations	-0.389463
Other	-0.404045
Poor.physical.health.days	-0.459504
Management.professional.and.related.occupations	-0.520295
Poverty.Rate.below.federal.poverty.threshold	-0.545999
Asian	-0.644730
Construction.extraction.maintenance.and.repair.occupations	-0.774297
Service.occupations	-1.087902
Median Age	-1.110052
Unemployment	-1.487912
Farming.fishing.and.forestry.occupations	-1.524990
Latitude	-1.758330
Graduate Degree	-2.226425

Below is a full list for Class 4:

```
Less Than High School Diploma                                  1.981712
Gini.Coefficient                                               1.912213
Amerindian                                                     1.236282
Adult.obesity                                                  1.222738
Sales.and.office.occupations                                   1.100736
Median Earnings 2010                                           0.945537
Production.transportation.and.material.moving.occupations      0.870096
Total Confirmed                                                0.636670
Uninsured                                                      0.533183
Adults 65 and Older Living in Poverty                          0.512948
Total Death                                                    0.348654
White                                                          0.272687
Hispanic                                                       0.255248
Sexually.transmitted.infections                                0.174738
At Least Bachelors's Degree                                    0.078208
Management.professional.and.related.occupations                0.041293
Fatality Rate                                                 -0.019297
Children Under 6 Living in Poverty                            -0.041299
Precincts                                                     -0.049511
Longitude                                                     -0.051894
Total Population                                              -0.069366
White  Asian                                                  -0.107528
Diabetes                                                      -0.148172
Other                                                         -0.195779
School Enrollment                                             -0.386630
Adult.smoking                                                 -0.565232
Service.occupations                                           -0.613431
Poor.physical.health.days                                     -0.616288
Asian                                                         -0.860795
Poverty.Rate.below.federal.poverty.threshold                  -0.974237
Farming.fishing.and.forestry.occupations                      -1.017664
Poor.mental.health.days                                       -1.216493
Graduate Degree                                               -1.217883
Black                                                         -1.260107
At Least High School Diploma                                  -1.294833
Latitude                                                      -1.453969
Construction.extraction.maintenance.and.repair.occupations    -1.504750
Unemployment                                                  -2.096750
Median Age                                                    -2.134841
dtype: float64
```

Below is a full list for Class 5:

Adults 65 and Older Living in Poverty	1.666582
Service.occupations	1.216860
Less Than High School Diploma	1.108252
Amerindian	0.926691
Sales.and.office.occupations	0.797626
Latitude	0.628143
Hispanic	0.607997
Gini.Coefficient	0.384383
Adult.obesity	0.355962
Total Confirmed	0.340895
Poverty.Rate.below.federal.poverty.threshold	0.268141
Total Death	0.112811
Sexually.transmitted.infections	0.012776
Longitude	0.009486
Total Population	-0.042247
Children Under 6 Living in Poverty	-0.054262
At Least Bachelors's Degree	-0.145760
Precincts	-0.224212
Production.transportation.and.material.moving.occupations	-0.241435
Management.professional.and.related.occupations	-0.270651
White	-0.334712
Uninsured	-0.356217
Other	-0.433021
Poor.physical.health.days	-0.434071
Asian	-0.457042
Unemployment	-0.466998
Farming.fishing.and.forestry.occupations	-0.518164
White Asian	-0.537104
Black	-0.702559
At Least High School Diploma	-0.722944
Fatality Rate	-0.926609
Construction.extraction.maintenance.and.repair.occupations	-0.927851
School Enrollment	-0.987048
Graduate Degree	-0.995721
Median Earnings 2010	-1.036271
Adult.smoking	-1.104938
Poor.mental.health.days	-1.314758
Diabetes	-1.378629
Median Age	-1.715972

Multiclass: Fatality Rate:

The testing accuracy for the 6-class classification with the fatality rate is 46.18%, and the training accuracy is 45.18%.

Confusion Matrix:

Actual \ Predicted	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	16	10	4	1	1
2	0	48	251	120	29	6
3	0	7	28	55	29	12
4	0	1	1	4	1	3
5	0	1	0	0	0	0

According to the multiclass confusion matrix, there are not many correctly predicted for all classes. Class 2 and Class 3 predicted the most, and Class 2 has the most correctly predicted number. There are also not many false-negatives with a significant difference.

Classification Report:

Based on the classification report, the highest precision score is in Class 2 with 87% with a slightly better recall score of 55% than the positivity rate. However, it is not as good as the binary classification. The recall score for most of the classes is still around 50% and for Class 4 is 10%. Overall, logistic regression does not perform well in multiclass with the fatality rate as well.

Classification Report

	precision	recall	f1-score	support
1	0.22	0.50	0.30	32
2	0.87	0.55	0.67	454
3	0.30	0.42	0.35	131
4	0.02	0.10	0.03	10
5	0.00	0.00	0.00	1
accuracy			0.51	628
macro avg	0.28	0.31	0.27	628
weighted avg	0.70	0.51	0.58	628

Weight on features: There is no class 0

As shown below, the common features with a high impact on the fatality rate for lower-risk classes are at least Bachelor's degree, and for higher-risk classes are the median age. However, the median age is, again, the opposite in the lower-risk classes for less impact. Occupations in farming, fishing, and forestry do not seem to have a high impact on all classes, except class 5. One thing to keep in mind is that there is only one data point for class 5 in our model. Therefore, class 5 is irrelevant in this model.

Below is a full list for Class 1

At Least Bachelors's Degree	1.871525
Latitude	1.636800
Construction.extraction.maintenance.and.repair.occupations	1.480989
Other	1.025152
Farming.fishing.and.forestry.occupations	1.019384
Amerindian	0.685525
White	0.656988
White Asian	0.521417
Graduate Degree	0.435325
Sales.and.office.occupations	0.298155
Poverty.Rate.below.federal.poverty.threshold	0.276111
Service.occupations	0.201152
Adult.smoking	0.140773
At Least High School Diploma	0.106700
Median Earnings 2010	0.040717
Poor.physical.health.days	0.007873
Unemployment	0.004231
Sexually.transmitted.infections	0.001795
Poor.mental.health.days	-0.005020
Management.professional.and.related.occupations	-0.161137
Total Confirmed	-0.245475
Asian	-0.308242
Uninsured	-0.358284
Black	-0.421464
Adults 65 and Older Living in Poverty	-0.596845
Total Population	-0.671279
Less Than High School Diploma	-0.727134
Precincts	-0.753234
Total Death	-0.791754
Positivity Rate	-0.822742
Adult.obesity	-1.102058
Production.transportation.and.material.moving.occupations	-1.176014
Hispanic	-1.353828
Children Under 6 Living in Poverty	-1.491941
Longitude	-1.579925
Gini.Coefficient	-1.594534
School Enrollment	-1.647798
Diabetes	-3.063332
Median Age	-3.384319
dtype: float64	

Below is a full list for Class 2:

Positivity Rate	1.602867
At Least Bachelors's Degree	1.277375
Sales.and.office.occupations	1.155093
Total Population	1.093476
Other	1.085285
Total Confirmed	0.907142
Graduate Degree	0.775388
Precincts	0.688303
Poor.physical.health.days	0.597269
Latitude	0.591870
Asian	0.578349
Adult.obesity	0.553296
Poor.mental.health.days	0.390334
At Least High School Diploma	0.374084
Unemployment	0.354913
Production.transportation.and.material.moving.occupations	0.328486
Less Than High School Diploma	0.295016
White Asian	0.283433
Adult.smoking	0.247675
Construction.extraction.maintenance.and.repair.occupations	0.126837
Children Under 6 Living in Poverty	0.119854
Median Earnings 2010	0.038659
White	0.027759
Poverty.Rate.below.federal.poverty.threshold	-0.013324
Hispanic	-0.197775
Longitude	-0.199617
Black	-0.326945
Management.professional.and.related.occupations	-0.401254
Service.occupations	-0.405435
Adults 65 and Older Living in Poverty	-0.422994
School Enrollment	-0.479003
Amerindian	-0.491454
Sexually.transmitted.infections	-0.600373
Uninsured	-0.669341
Diabetes	-0.707663
Total Death	-0.757198
Gini.Coefficient	-0.796484
Farming.fishing.and.forestry.occupations	-0.891654
Median Age	-2.020037

Below is a full list for Class 3:

Longitude	1.598889
School Enrollment	1.451177
Adult.obesity	1.066607
Children Under 6 Living in Poverty	1.004965
Median Earnings 2010	0.969888
Hispanic	0.941537
Production.transportation.and.material.moving.occupations	0.837442
Gini.Coefficient	0.781154
Diabetes	0.730469
Poor.mental.health.days	0.713608
Median Age	0.713093
Positivity Rate	0.682883
Graduate Degree	0.638476
Sales.and.office.occupations	0.560679
Total Death	0.524987
Sexually.transmitted.infections	0.417341
Unemployment	0.249160
Adult.smoking	0.120809
Service.occupations	-0.020714
Adults 65 and Older Living in Poverty	-0.043346
Precincts	-0.087106
Less Than High School Diploma	-0.089748
Poor.physical.health.days	-0.092710
White	-0.119247
Asian	-0.126754
White Asian	-0.175409
Amerindian	-0.197337
Management.professional.and.related.occupations	-0.287655
At Least High School Diploma	-0.299116
Total Population	-0.361973
Other	-0.362795
Black	-0.410801
Total Confirmed	-0.501384
Uninsured	-0.537357
At Least Bachelors's Degree	-0.703957
Poverty.Rate.below.federal.poverty.threshold	-0.868008
Construction.extraction.maintenance.and.repair.occupations	-0.935829
Farming.fishing.and.forestry.occupations	-1.064814
Latitude	-1.115640

dtype: float64

Below is a full list for Class 4:

Median Age	2.091437
Diabetes	1.710271
Uninsured	1.130571
Total Death	1.111333
School Enrollment	0.805903
Management.professional.and.related.occupations	0.573122
Adults 65 and Older Living in Poverty	0.529194
Gini.Coefficient	0.476205
Hispanic	0.431447
Less Than High School Diploma	0.417880
Poverty.Rate.below.federal.poverty.threshold	0.397367
Black	0.364139
Precincts	0.339965
Amerindian	0.300575
Asian	0.266551
Production.transportation.and.material.moving.occupations	0.220767
Total Population	0.126460
Longitude	0.113995
Adult.obesity	0.090726
Sexually.transmitted.infections	0.090231
At Least High School Diploma	0.037818
Total Confirmed	0.009166
Children Under 6 Living in Poverty	-0.031746
Poor.physical.health.days	-0.072613
Service.occupations	-0.105789
Sales.and.office.occupations	-0.112653
Median Earnings 2010	-0.123945
Positivity Rate	-0.236912
Unemployment	-0.298060
White Asian	-0.318385
White	-0.435757
Construction.extraction.maintenance.and.repair.occupations	-0.543047
Poor.mental.health.days	-0.544421
Adult.smoking	-0.555995
Latitude	-0.560290
Graduate Degree	-0.784165
Farming.fishing.and.forestry.occupations	-0.814835
Other	-1.210379
At Least Bachelors's Degree	-1.382927

Below is a full list for Class 5:

Median Age	2.599825
Farming.fishing.and.forestry.occupations	1.751919
Diabetes	1.330255
Gini.Coefficient	1.133660
Black	0.795071
Adults 65 and Older Living in Poverty	0.533990
Uninsured	0.434411
Children Under 6 Living in Poverty	0.398867
Service.occupations	0.330786
Management.professional.and.related.occupations	0.276923
Poverty.Rate.below.federal.poverty.threshold	0.207854
Hispanic	0.178619
Less Than High School Diploma	0.103986
Sexually.transmitted.infections	0.091006
Longitude	0.066658
Adult.smoking	0.046738
Total Death	-0.087369
Construction.extraction.maintenance.and.repair.occupations	-0.128949
White	-0.129743
School Enrollment	-0.130279
Total Confirmed	-0.169450
Total Population	-0.186684
Precincts	-0.187929
Production.transportation.and.material.moving.occupations	-0.210680
At Least High School Diploma	-0.219486
Amerindian	-0.297309
Unemployment	-0.310243
White Asian	-0.311056
Asian	-0.409904
Poor.physical.health.days	-0.439820
Other	-0.537263
Latitude	-0.552740
Poor.mental.health.days	-0.554501
Adult.obesity	-0.608571
Median Earnings 2010	-0.925319
At Least Bachelors's Degree	-1.062016
Graduate Degree	-1.065024
Positivity Rate	-1.226096
Sales.and.office.occupations	-1.901275

Summary:
As a result, logistic regression in multiclass does not perform well for our dataset because it is a very simple algorithm.

	Testing accuracy	Training accuracy
Binary Positivity Rate	70.38%	70.10%
Binary Fatality Rate	71.34%	68.65%
Multiclass Positivity Rate	49.52%	51.51%
Multiclass Fatality Rate	51.43%	51.99%

- **SVM**

- In using a Support Vector Machine model, we figured that for the classification it would perform very well. To achieve our SVM model, we used a linear kernel for the model on both binary and multiclass classifications of positivity rate and fatality rate. The linear kernel was chosen because it took into account the fact that our data had a very large number of features and if we were to use a kernel made for nonlinear data like a radial basis function kernel or a polynomial kernel, we could give the model an opportunity to overfit our data. All the iterations of the model were given a split of 80% training data and 20% testing data.

Positivity Rate (Binary Classification) :

```
Accuracy:  70.70 %
              precision    recall  f1-score   support

         0.0       0.73      0.68      0.70       320
         1.0       0.69      0.74      0.71       308

   micro avg       0.71      0.71      0.71       628
   macro avg       0.71      0.71      0.71       628
weighted avg       0.71      0.71      0.71       628
```

Classification Report Results:

- The accuracy shows that our binary classification of the fatality rate performed at 70.70% (with a training accuracy of 70.74%) which is almost as good as the accuracy we saw on the binary

classification of positivity rate. The precision shows that our model is able to have a very low false positive rate in the case of both classes and identifies both the 0 and 1 class at 0.73 and 0.69 respectively which are good numbers in how correct it can be for correct identification. The recall shows that in identifying true positive rates, the model performed well on both classes but slightly better on class 1. The F1 scores for the two classes show that similarly the model performed well on both classes but slightly better on class 1. Overall, the model performed well on both of the classes in binary classification which is good.

Positivity Rate (Multiclass Classification) :

```
Accuracy:  49.20 %
              precision    recall  f1-score   support

           0       0.40      0.11      0.17        18
           1       0.50      0.03      0.06        60
           2       0.50      0.58      0.54       242
           3       0.49      0.71      0.58       230
           4       0.00      0.00      0.00        53
           5       0.00      0.00      0.00        25

   micro avg       0.49      0.49      0.49       628
   macro avg       0.31      0.24      0.23       628
weighted avg       0.43      0.49      0.43       628
```

Classification Report Results:

- The accuracy shows that our multiclass classification of the positivity rate performed at 49.20% (with a training accuracy of 50.84%) which illustrates that the model was able to classify almost half of the data given when introduced to 6 classes instead of 2. The precision shows that our model is able to only identify 4 of our classes which are 0, 1, 2 and 3 but almost disregards classes 4 and 5 which are some of the higher and more critical classes that need to be identified. The recall shows that in identifying true positives, the model only identifies classes 0, 1, 2 and 3 like the precision but only performed well on classes 2 and 3 which is not good for the purpose of our model at all. The F1 scores for the multiple classes show that similarly only classes 0, 1, 2 and 3 were identified and performed well on only classes 2 and 3. Overall, the model performed terribly on the class for multiclass classification in terms of recall and f1 score and more importantly it didn't even identify the two most critical levels of positivity which is not ideal for the model at all if we want to correctly identify and save people from testing positive from the virus.

Fatality Rate (Binary Classification):

```
Accuracy:  69.11 %
              precision    recall  f1-score   support

         0.0       0.71      0.80      0.75       363
         1.0       0.66      0.55      0.60       265

   micro avg       0.69      0.69      0.69       628
   macro avg       0.68      0.67      0.67       628
weighted avg       0.69      0.69      0.69       628
```

Classification Report Results:

- The accuracy shows that our binary classification of the fatality rate performed at 69.11% (with a training accuracy of 68.87%) which is almost as good as the accuracy we saw on the binary classification of positivity rate. The precision shows that our model is able to have a very low false positive rate in the case of both classes and identifies both the 0 and 1 class at 0.71 and 0.66 respectively which are good numbers in how correct it can be for correct identification. The recall shows that in identifying true positive rates, the model performed well on both classes but much better on class 0. The F1 scores for the two classes show that similarly the model performed well on both classes but much better on class 0. Overall, the model performed well on both classes but much better on class 0 which makes sense as it'd be easier to identify and classify someone in the 0 class than the 1 class.

Fatality Rate (Multiclass Classification)

```
Accuracy:  51.43 %
              precision    recall  f1-score   support

           1       0.63      0.16      0.26        73
           2       0.53      0.92      0.68       290
           3       0.41      0.24      0.30       183
           4       0.00      0.00      0.00        60
           5       0.00      0.00      0.00        22

   micro avg       0.51      0.51      0.51       628
   macro avg       0.31      0.27      0.25       628
weighted avg       0.44      0.51      0.43       628
```

Classification Report Results:

- The accuracy shows that our multiclass classification of the fatality rate performed at 51.43% (with a training accuracy of 51.83%) which illustrates that the model was able to classify almost half of the data given when introduced to 6 classes instead of 2. The precision shows that our model is able to only identify 3 of our classes which are 1, 2 and 3 but disregards classes 4 and 5

which are some of the higher and more critical classes that need to be identified. The recall shows that in identifying true positives, the model only identifies classes 1, 2 and 3 like the precision but only performed well on class 2 which is not good for the purpose of our model at all. The F1 scores for the multiple classes show that similarly only classes 1, 2 and 3 were identified and the model performed well on only class 2. Overall, the model performed terribly on the class for multiclass classification in terms of recall and f1 score and more importantly it didn't even identify the two most critical levels of fatality which is not ideal for the model at all if we want to correctly identify and save people from testing positive from the virus.

Overall SVM Model Results:

- In starting this model, we assumed that the SVM model would perform considerably well on the data. Based on our results, however, the model performed decently well. While we expected higher results, this could be due to multiple factors such as issues with the data, the model's parameters or even the chosen target variables of positivity rate or fatality rate. An intended hypothesis on why the model had such low results across the board was that some of the features being passed to the data were irrelevant features. While the SVM model, especially the linear kernel version of the model, can handle large sets of features on data if there is a significant amount of irrelevant features within the data this can skew results and cause disparaging accuracy numbers compared to a model with the best chosen features. As a result, of our model we can see in comparison of both binary and multiclass classification that the positivity rate was definitely the more superior target value than the fatality rate regardless of if it was binary or multiclass at least in the case of the SVM model.

- **Neural Network**

The train and test dataset has been splitted with 80/20 ratio. 80% is train dataset and 20% is test dataset. During training, we also split the train again with 80/20 ratio to get 80% of data as train and 20% data as validation dataset. We also explored using the positivity rate and fatality rate as different indicators of our risk labeling.

We first start with three layers model, 40 nodes in the first layer since we have 40 initial inputs, 80 nodes in the second layer, 40 nodes in the second layer, and the last layer to produce classification results.

1. Binary Classification

```python
# Neural network
model = Sequential()
model.add(Dense(40, activation='relu'))
model.add(Dense(80, activation='relu'))
model.add(Dense(40,activation='relu'))
#model.add(Dense(256, activation='relu'))
#model.add(Dense(128,activation='relu'))
#model.add(Dense(32,activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
#model.add(Dense(len(class_names),activation='softmax'))
#model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['acc'])
#model.summary()
history = model.fit(X_train, y_train, validation_split=0.2,epochs=25, batch_size=64)
y_pred = model.predict(X_test)
```

- Positivity Rate:

```
Epoch 1/25
32/32 [==============================] - 1s 19ms/step - loss: 0.6895 - accuracy: 0.5416 - val_loss: 0.6793 - val_accuracy: 0.5726
Epoch 2/25
32/32 [==============================] - 0s 3ms/step - loss: 0.6800 - accuracy: 0.5602 - val_loss: 0.6650 - val_accuracy: 0.5785
Epoch 3/25
32/32 [==============================] - 0s 3ms/step - loss: 0.6594 - accuracy: 0.6240 - val_loss: 0.6530 - val_accuracy: 0.5865
Epoch 4/25
32/32 [==============================] - 0s 3ms/step - loss: 0.6490 - accuracy: 0.6119 - val_loss: 0.6204 - val_accuracy: 0.6799
Epoch 5/25
32/32 [==============================] - 0s 2ms/step - loss: 0.6237 - accuracy: 0.6624 - val_loss: 0.6269 - val_accuracy: 0.6402
Epoch 6/25
32/32 [==============================] - 0s 2ms/step - loss: 0.6021 - accuracy: 0.6776 - val_loss: 0.5839 - val_accuracy: 0.7117
Epoch 7/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5893 - accuracy: 0.6851 - val_loss: 0.5764 - val_accuracy: 0.7058
Epoch 8/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5841 - accuracy: 0.6891 - val_loss: 0.5701 - val_accuracy: 0.6859
Epoch 9/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5670 - accuracy: 0.6944 - val_loss: 0.5577 - val_accuracy: 0.7237
Epoch 10/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5699 - accuracy: 0.7043 - val_loss: 0.5493 - val_accuracy: 0.7276
Epoch 11/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5860 - accuracy: 0.6974 - val_loss: 0.5677 - val_accuracy: 0.7018
Epoch 12/25
32/32 [==============================] - 0s 2ms/step - loss: 0.5822 - accuracy: 0.6735 - val_loss: 0.5523 - val_accuracy: 0.7058
Epoch 13/25
32/32 [==============================] - 0s 2ms/step - loss: 0.5557 - accuracy: 0.7132 - val_loss: 0.5543 - val_accuracy: 0.7296
Epoch 14/25
32/32 [==============================] - 0s 2ms/step - loss: 0.5379 - accuracy: 0.7323 - val_loss: 0.5488 - val_accuracy: 0.7356
Epoch 15/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5329 - accuracy: 0.7440 - val_loss: 0.5402 - val_accuracy: 0.7256
Epoch 16/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5458 - accuracy: 0.7261 - val_loss: 0.5346 - val_accuracy: 0.7336
Epoch 17/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5464 - accuracy: 0.7153 - val_loss: 0.5283 - val_accuracy: 0.7435
Epoch 18/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5197 - accuracy: 0.7539 - val_loss: 0.5236 - val_accuracy: 0.7475
Epoch 19/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5420 - accuracy: 0.7114 - val_loss: 0.5229 - val_accuracy: 0.7455
Epoch 20/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5310 - accuracy: 0.7367 - val_loss: 0.5375 - val_accuracy: 0.7296
Epoch 21/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5415 - accuracy: 0.7152 - val_loss: 0.5789 - val_accuracy: 0.6978
Epoch 22/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5484 - accuracy: 0.7278 - val_loss: 0.5196 - val_accuracy: 0.7435
Epoch 23/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5240 - accuracy: 0.7464 - val_loss: 0.5407 - val_accuracy: 0.7137
Epoch 24/25
32/32 [==============================] - 0s 2ms/step - loss: 0.5314 - accuracy: 0.7304 - val_loss: 0.5112 - val_accuracy: 0.7455
Epoch 25/25
32/32 [==============================] - 0s 2ms/step - loss: 0.5073 - accuracy: 0.7520 - val_loss: 0.5127 - val_accuracy: 0.7455
Accuracy is: 72.92993630573248
```

```
              precision    recall  f1-score   support

         0.0       0.74      0.69      0.71       306
         1.0       0.72      0.77      0.74       322

    accuracy                           0.73       628
   macro avg       0.73      0.73      0.73       628
weighted avg       0.73      0.73      0.73       628
```

From the above results, we can see both train and validation losses are decreasing with more epochs as we expected. Our NN model achieves 73% f1-score with 25 epochs. And the accuracy is improving with more epochs, but not significantly. So, we take the results with 25 epochs.

Classification report analysis:
Recall score shows what percent of the positive cases did you catch. And precision score shows what percent of your predictions were correct. We can see that our model performs relatively better on high risk class detection than another class detection.

As we can see, the results above are not good enough. To achieve better performance, we increased hidden layers and hidden nodes in each layer to make the model to be more advanced than the previous model.

Advanced model:

```
# Neural network
model = Sequential()
model.add(Dense(128, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(512,activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(128,activation='relu'))
model.add(Dense(32,activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
#model.add(Dense(len(class_names),activation='softmax'))
#model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['acc'])
#model.summary()
history = model.fit(X_train, y_train, validation_split=0.2,epochs=25, batch_size=64)
y_pred = model.predict(X_test)
```

```
Epoch 1/25
32/32 [==============================] - 2s 29ms/step - loss: 0.6962 - accuracy: 0.5273 - val_loss: 0.6846 - val_accuracy: 0.5288
Epoch 2/25
32/32 [==============================] - 0s 4ms/step - loss: 0.6513 - accuracy: 0.6185 - val_loss: 0.6222 - val_accuracy: 0.6362
Epoch 3/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5976 - accuracy: 0.6826 - val_loss: 0.6785 - val_accuracy: 0.6382
Epoch 4/25
32/32 [==============================] - 0s 4ms/step - loss: 0.6695 - accuracy: 0.6225 - val_loss: 0.6274 - val_accuracy: 0.6461
Epoch 5/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5919 - accuracy: 0.7020 - val_loss: 0.6271 - val_accuracy: 0.6302
Epoch 6/25
32/32 [==============================] - 0s 4ms/step - loss: 0.6128 - accuracy: 0.6818 - val_loss: 0.6033 - val_accuracy: 0.6640
Epoch 7/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5707 - accuracy: 0.7056 - val_loss: 0.6285 - val_accuracy: 0.6581
Epoch 8/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5512 - accuracy: 0.7108 - val_loss: 0.6076 - val_accuracy: 0.6600
Epoch 9/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5555 - accuracy: 0.7028 - val_loss: 0.5802 - val_accuracy: 0.6819
Epoch 10/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5534 - accuracy: 0.7004 - val_loss: 0.5978 - val_accuracy: 0.6819
Epoch 11/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5145 - accuracy: 0.7407 - val_loss: 0.6514 - val_accuracy: 0.6541
Epoch 12/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5533 - accuracy: 0.7080 - val_loss: 0.5943 - val_accuracy: 0.6839
Epoch 13/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5202 - accuracy: 0.7308 - val_loss: 0.7068 - val_accuracy: 0.6561
Epoch 14/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5215 - accuracy: 0.7440 - val_loss: 0.5852 - val_accuracy: 0.6759
Epoch 15/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5292 - accuracy: 0.7321 - val_loss: 0.5852 - val_accuracy: 0.6819
Epoch 16/25
32/32 [==============================] - 0s 5ms/step - loss: 0.4976 - accuracy: 0.7347 - val_loss: 0.5907 - val_accuracy: 0.6978
Epoch 17/25
32/32 [==============================] - 0s 4ms/step - loss: 0.4738 - accuracy: 0.7720 - val_loss: 0.6163 - val_accuracy: 0.6799
Epoch 18/25
32/32 [==============================] - 0s 4ms/step - loss: 0.4749 - accuracy: 0.7596 - val_loss: 0.5750 - val_accuracy: 0.6700
Epoch 19/25
32/32 [==============================] - 0s 5ms/step - loss: 0.4894 - accuracy: 0.7481 - val_loss: 0.6101 - val_accuracy: 0.6899
Epoch 20/25
32/32 [==============================] - 0s 5ms/step - loss: 0.4639 - accuracy: 0.7713 - val_loss: 0.5749 - val_accuracy: 0.6938
Epoch 21/25
32/32 [==============================] - 0s 3ms/step - loss: 0.4738 - accuracy: 0.7642 - val_loss: 0.5956 - val_accuracy: 0.6998
Epoch 22/25
32/32 [==============================] - 0s 4ms/step - loss: 0.4563 - accuracy: 0.7704 - val_loss: 0.6243 - val_accuracy: 0.6998
Epoch 23/25
32/32 [==============================] - 0s 4ms/step - loss: 0.4366 - accuracy: 0.7817 - val_loss: 0.6308 - val_accuracy: 0.6879
Epoch 24/25
32/32 [==============================] - 0s 5ms/step - loss: 0.4677 - accuracy: 0.7524 - val_loss: 0.5883 - val_accuracy: 0.6879
Epoch 25/25
32/32 [==============================] - 0s 4ms/step - loss: 0.4073 - accuracy: 0.8026 - val_loss: 0.5880 - val_accuracy: 0.7097
Accuracy is: 75.95541401273886
```

	precision	recall	f1-score	support
0.0	0.73	0.76	0.74	287
1.0	0.79	0.76	0.77	341
accuracy			0.76	628
macro avg	0.76	0.76	0.76	628
weighted avg	0.76	0.76	0.76	628

From the above results, we can see both train and validation losses are decreasing with more epochs as expected. Our NN model achieves 76% f1-score with 25 epochs. The accuracy is improving with more epochs, but not significantly. So, we take the results with 25 epochs.

Classification report analysis:
We can see that our model works well on both low and high risk classes detection.

The more advanced NN model achieves a better f1-score, but not significantly. So, we'd like to use this model instead of having a more advanced model.

In our experiments, we also found that increasing the epochs would increase train accuracy but decrease validation decreasing, which might indicate the model is overfitting. For example, the below screenshot shows train and validation accuracy with 200 epochs, we can see that the train accuray kept going up to 0.91, but the validation accuracy stays around 71%.

```
Epoch 191/200
32/32 [==============================] - 0s 2ms/step - loss: 0.2256 - accuracy: 0.9074 - val_loss: 1.1244 - val_accuracy: 0.7237
Epoch 192/200
32/32 [==============================] - 0s 2ms/step - loss: 0.2200 - accuracy: 0.9081 - val_loss: 1.0617 - val_accuracy: 0.7475
Epoch 193/200
32/32 [==============================] - 0s 2ms/step - loss: 0.2103 - accuracy: 0.9113 - val_loss: 1.1385 - val_accuracy: 0.7316
Epoch 194/200
32/32 [==============================] - 0s 3ms/step - loss: 0.2186 - accuracy: 0.9011 - val_loss: 1.1173 - val_accuracy: 0.7137
Epoch 195/200
32/32 [==============================] - 0s 2ms/step - loss: 0.2087 - accuracy: 0.9137 - val_loss: 1.1101 - val_accuracy: 0.7117
Epoch 196/200
32/32 [==============================] - 0s 2ms/step - loss: 0.1928 - accuracy: 0.9197 - val_loss: 1.1065 - val_accuracy: 0.7296
Epoch 197/200
32/32 [==============================] - 0s 2ms/step - loss: 0.2016 - accuracy: 0.9187 - val_loss: 1.1300 - val_accuracy: 0.7376
Epoch 198/200
32/32 [==============================] - 0s 3ms/step - loss: 0.1959 - accuracy: 0.9305 - val_loss: 1.1435 - val_accuracy: 0.7237
Epoch 199/200
32/32 [==============================] - 0s 3ms/step - loss: 0.2237 - accuracy: 0.9055 - val_loss: 1.1570 - val_accuracy: 0.7177
Epoch 200/200
32/32 [==============================] - 0s 2ms/step - loss: 0.2129 - accuracy: 0.9150 - val_loss: 1.2132 - val_accuracy: 0.7197
Accuracy is: 73.56687898089172
              precision    recall  f1-score   support

         0.0       0.76      0.66      0.70       301
         1.0       0.72      0.81      0.76       327

    accuracy                           0.74       628
   macro avg       0.74      0.73      0.73       628
weighted avg       0.74      0.74      0.73       628
```

To resolve this problem, we add a dropout layer after each hidden layer. We can see that both train and validation accuracy are around 75%. We fix the overfitting issue, however, the test results are not getting too much improvement. So, adding dropout layers doesn't really improve the model performance.

```python
# Neural network
model = Sequential()
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(32,activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
#model.add(Dense(len(class_names),activation='softmax'))
#model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['acc'])
#model.summary()
history = model.fit(X_train, y_train, validation_split=0.2,epochs=25, batch_size=64)
y_pred = model.predict(X_test)
```

```
32/32 [==============================] - 0s 5ms/step - loss: 0.4765 - accuracy: 0.7712 - val_loss: 0.5314 - val_accuracy: 0.7097
Epoch 193/200
32/32 [==============================] - 0s 4ms/step - loss: 0.4715 - accuracy: 0.7728 - val_loss: 0.5279 - val_accuracy: 0.7356
Epoch 194/200
32/32 [==============================] - 0s 4ms/step - loss: 0.4767 - accuracy: 0.7552 - val_loss: 0.5180 - val_accuracy: 0.7376
Epoch 195/200
32/32 [==============================] - 0s 5ms/step - loss: 0.4814 - accuracy: 0.7517 - val_loss: 0.5211 - val_accuracy: 0.7276
Epoch 196/200
32/32 [==============================] - 0s 4ms/step - loss: 0.5020 - accuracy: 0.7475 - val_loss: 0.5103 - val_accuracy: 0.7515
Epoch 197/200
32/32 [==============================] - 0s 4ms/step - loss: 0.4617 - accuracy: 0.7781 - val_loss: 0.5125 - val_accuracy: 0.7515
Epoch 198/200
32/32 [==============================] - 0s 5ms/step - loss: 0.4888 - accuracy: 0.7431 - val_loss: 0.5196 - val_accuracy: 0.7435
Epoch 199/200
32/32 [==============================] - 0s 5ms/step - loss: 0.5149 - accuracy: 0.7566 - val_loss: 0.5425 - val_accuracy: 0.7197
Epoch 200/200
32/32 [==============================] - 0s 5ms/step - loss: 0.4801 - accuracy: 0.7578 - val_loss: 0.5139 - val_accuracy: 0.7455
Accuracy is: 74.04458598726114
              precision    recall  f1-score   support

         0.0       0.78      0.67      0.72       316
         1.0       0.71      0.81      0.76       312

    accuracy                           0.74       628
   macro avg       0.75      0.74      0.74       628
weighted avg       0.75      0.74      0.74       628
```

- Fatality Rate :

```
Epoch 1/25
32/32 [==============================] - 1s 27ms/step - loss: 0.6669 - accuracy: 0.6024 - val_loss: 0.6392 - val_accuracy: 0.6262
Epoch 2/25
32/32 [==============================] - 0s 5ms/step - loss: 0.6237 - accuracy: 0.6455 - val_loss: 0.6041 - val_accuracy: 0.6700
Epoch 3/25
32/32 [==============================] - 0s 4ms/step - loss: 0.6048 - accuracy: 0.6756 - val_loss: 0.6026 - val_accuracy: 0.6660
Epoch 4/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5911 - accuracy: 0.6660 - val_loss: 0.6061 - val_accuracy: 0.6441
Epoch 5/25
32/32 [==============================] - 0s 3ms/step - loss: 0.5991 - accuracy: 0.6662 - val_loss: 0.5837 - val_accuracy: 0.6620
Epoch 6/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5669 - accuracy: 0.7020 - val_loss: 0.5780 - val_accuracy: 0.6700
Epoch 7/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5552 - accuracy: 0.7128 - val_loss: 0.5665 - val_accuracy: 0.6700
Epoch 8/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5811 - accuracy: 0.6857 - val_loss: 0.5643 - val_accuracy: 0.6819
Epoch 9/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5522 - accuracy: 0.6797 - val_loss: 0.6224 - val_accuracy: 0.6481
Epoch 10/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5714 - accuracy: 0.6733 - val_loss: 0.5648 - val_accuracy: 0.6958
Epoch 11/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5257 - accuracy: 0.7250 - val_loss: 0.5607 - val_accuracy: 0.6859
Epoch 12/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5295 - accuracy: 0.7187 - val_loss: 0.5576 - val_accuracy: 0.6700
Epoch 13/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5458 - accuracy: 0.7005 - val_loss: 0.5492 - val_accuracy: 0.6799
Epoch 14/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5323 - accuracy: 0.7212 - val_loss: 0.5576 - val_accuracy: 0.6680
Epoch 15/25
32/32 [==============================] - 0s 6ms/step - loss: 0.5244 - accuracy: 0.7122 - val_loss: 0.5567 - val_accuracy: 0.6938
Epoch 16/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5141 - accuracy: 0.7197 - val_loss: 0.5599 - val_accuracy: 0.6779
Epoch 17/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5208 - accuracy: 0.7168 - val_loss: 0.5320 - val_accuracy: 0.7117
Epoch 18/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5042 - accuracy: 0.7362 - val_loss: 0.5807 - val_accuracy: 0.6958
Epoch 19/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5033 - accuracy: 0.7448 - val_loss: 0.5496 - val_accuracy: 0.7276
Epoch 20/25
32/32 [==============================] - 0s 5ms/step - loss: 0.5050 - accuracy: 0.7357 - val_loss: 0.5705 - val_accuracy: 0.6839
Epoch 21/25
32/32 [==============================] - 0s 4ms/step - loss: 0.4980 - accuracy: 0.7334 - val_loss: 0.5511 - val_accuracy: 0.7018
Epoch 22/25
32/32 [==============================] - 0s 4ms/step - loss: 0.4759 - accuracy: 0.7611 - val_loss: 0.5383 - val_accuracy: 0.6859
Epoch 23/25
32/32 [==============================] - 0s 5ms/step - loss: 0.4765 - accuracy: 0.7487 - val_loss: 0.5616 - val_accuracy: 0.7137
Epoch 24/25
32/32 [==============================] - 0s 5ms/step - loss: 0.4815 - accuracy: 0.7500 - val_loss: 0.5729 - val_accuracy: 0.6759
Epoch 25/25
32/32 [==============================] - 0s 4ms/step - loss: 0.5198 - accuracy: 0.7068 - val_loss: 0.5398 - val_accuracy: 0.7157
Accuracy is: 72.77070063694268
```

```
              precision    recall  f1-score   support

         0.0       0.74      0.82      0.78       365
         1.0       0.71      0.59      0.65       263

    accuracy                           0.73       628
   macro avg       0.72      0.71      0.71       628
weighted avg       0.73      0.73      0.72       628
```

Comparing results from using positivity rate and fatality rate, we can see that positivity rate works better on binary classification. Using fatality rate, our model doesn't perform well on capturing high risk classes.

2. Multiclass classification:

We use how many standard deviations away from the mean as the measurement to classify our risk level into six different risk levels: 0,1,2,3,4,5. 0 is the safest, 5 is the highest risk class.

We start with the advanced model that we get from above binary classification since our task is getting more complicated. We also explored different parameters on the model to see whether our model can perform better. For example, we change the six layers NN model to four layers NN model. We also replace sigmoid with softmax for multiclass classification tasks.

Model:

```
# Neural network
model = Sequential()
model.add(Dense(256, activation='relu'))
model.add(Dense(512,activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(128,activation='relu'))
#model.add(Dense(32,activation='relu'))
#model.add(Dense(1, activation='sigmoid'))
#model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.add(Dense(len(class_names),activation='softmax'))
model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['acc'])
#model.summary()
history = model.fit(X_train, y_train, validation_split=0.2,epochs=90, batch_size=64)
y_pred = model.predict(X_test)
```

- Positivity Rate

```
Epoch 80/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2560 - acc: 0.8883 - val_loss: 1.9614 - val_acc: 0.5626
Epoch 81/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2281 - acc: 0.9046 - val_loss: 1.8738 - val_acc: 0.5527
Epoch 82/90
32/32 [==============================] - 0s 4ms/step - loss: 0.1801 - acc: 0.9458 - val_loss: 2.0200 - val_acc: 0.5388
Epoch 83/90
32/32 [==============================] - 0s 4ms/step - loss: 0.1886 - acc: 0.9292 - val_loss: 2.1036 - val_acc: 0.5547
Epoch 84/90
32/32 [==============================] - 0s 4ms/step - loss: 0.1337 - acc: 0.9541 - val_loss: 2.2305 - val_acc: 0.5408
Epoch 85/90
32/32 [==============================] - 0s 4ms/step - loss: 0.1386 - acc: 0.9500 - val_loss: 2.2880 - val_acc: 0.5527
Epoch 86/90
32/32 [==============================] - 0s 4ms/step - loss: 0.1983 - acc: 0.9189 - val_loss: 2.1539 - val_acc: 0.5288
Epoch 87/90
32/32 [==============================] - 0s 4ms/step - loss: 0.1794 - acc: 0.9319 - val_loss: 2.1128 - val_acc: 0.5706
Epoch 88/90
32/32 [==============================] - 0s 5ms/step - loss: 0.1529 - acc: 0.9428 - val_loss: 2.2598 - val_acc: 0.5229
Epoch 89/90
32/32 [==============================] - 0s 4ms/step - loss: 0.1929 - acc: 0.9191 - val_loss: 2.3683 - val_acc: 0.5388
Epoch 90/90
32/32 [==============================] - 0s 3ms/step - loss: 0.1486 - acc: 0.9505 - val_loss: 2.2806 - val_acc: 0.5487
Accuracy is: 53.503184713375795
```

	precision	recall	f1-score	support
0	0.64	0.72	0.68	25
1	0.43	0.55	0.48	53
2	0.55	0.55	0.55	224
3	0.56	0.57	0.56	243
4	0.42	0.29	0.34	68
5	0.60	0.40	0.48	15
accuracy			0.54	628
macro avg	0.53	0.51	0.52	628
weighted avg	0.53	0.54	0.53	628

Classification report analysis:
We can see that our model is performing relatively better on risk class 2 and class 3 detection, achieving 0.55 and 0.56 f1-score.

From the above results, we can see both train accuracy is improving with more epochs, but validation accuracy is jumping around and stays around 53%. The nonincreasing validation accuracy might indicate the model overfitting. Our NN model achieves 0.54 f1-score with 90 epochs.

- Fatality Rate:

```
32/32 [==============================] - 0s 4ms/step - loss: 0.4829 - acc: 0.8231 - val_loss: 1.6315 - val_acc: 0.5408
Epoch 65/90
32/32 [==============================] - 0s 4ms/step - loss: 0.4758 - acc: 0.8081 - val_loss: 1.6578 - val_acc: 0.5388
Epoch 66/90
32/32 [==============================] - 0s 4ms/step - loss: 0.4645 - acc: 0.8190 - val_loss: 1.5941 - val_acc: 0.5427
Epoch 67/90
32/32 [==============================] - 0s 4ms/step - loss: 0.4385 - acc: 0.8261 - val_loss: 1.6858 - val_acc: 0.5328
Epoch 68/90
32/32 [==============================] - 0s 4ms/step - loss: 0.4580 - acc: 0.8157 - val_loss: 1.7167 - val_acc: 0.5427
Epoch 69/90
32/32 [==============================] - 0s 4ms/step - loss: 0.4563 - acc: 0.8140 - val_loss: 1.8060 - val_acc: 0.5626
Epoch 70/90
32/32 [==============================] - 0s 3ms/step - loss: 0.4292 - acc: 0.8303 - val_loss: 1.7313 - val_acc: 0.5527
Epoch 71/90
32/32 [==============================] - 0s 4ms/step - loss: 0.3823 - acc: 0.8532 - val_loss: 1.8170 - val_acc: 0.5547
Epoch 72/90
32/32 [==============================] - 0s 4ms/step - loss: 0.3929 - acc: 0.8516 - val_loss: 1.8583 - val_acc: 0.5388
Epoch 73/90
32/32 [==============================] - 0s 4ms/step - loss: 0.3597 - acc: 0.8675 - val_loss: 1.9026 - val_acc: 0.5487
Epoch 74/90
32/32 [==============================] - 0s 3ms/step - loss: 0.3568 - acc: 0.8573 - val_loss: 1.9802 - val_acc: 0.5209
Epoch 75/90
32/32 [==============================] - 0s 4ms/step - loss: 0.3565 - acc: 0.8648 - val_loss: 1.9464 - val_acc: 0.5408
Epoch 76/90
32/32 [==============================] - 0s 4ms/step - loss: 0.3960 - acc: 0.8411 - val_loss: 1.9653 - val_acc: 0.5467
Epoch 77/90
32/32 [==============================] - 0s 4ms/step - loss: 0.3277 - acc: 0.8789 - val_loss: 1.9998 - val_acc: 0.5209
Epoch 78/90
32/32 [==============================] - 0s 5ms/step - loss: 0.3055 - acc: 0.8932 - val_loss: 2.0358 - val_acc: 0.5169
Epoch 79/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2835 - acc: 0.8952 - val_loss: 2.1911 - val_acc: 0.5308
Epoch 80/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2833 - acc: 0.8895 - val_loss: 2.2992 - val_acc: 0.5467
Epoch 81/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2454 - acc: 0.9032 - val_loss: 2.3826 - val_acc: 0.5308
Epoch 82/90
32/32 [==============================] - 0s 3ms/step - loss: 0.2431 - acc: 0.9038 - val_loss: 2.4294 - val_acc: 0.4990
Epoch 83/90
32/32 [==============================] - 0s 3ms/step - loss: 0.2640 - acc: 0.9012 - val_loss: 2.3285 - val_acc: 0.5209
Epoch 84/90
32/32 [==============================] - 0s 3ms/step - loss: 0.2242 - acc: 0.9166 - val_loss: 2.4635 - val_acc: 0.5129
Epoch 85/90
32/32 [==============================] - 0s 5ms/step - loss: 0.2809 - acc: 0.8860 - val_loss: 2.2934 - val_acc: 0.5467
Epoch 86/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2913 - acc: 0.8907 - val_loss: 2.4805 - val_acc: 0.5209
Epoch 87/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2354 - acc: 0.9127 - val_loss: 2.4209 - val_acc: 0.5268
Epoch 88/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2175 - acc: 0.9344 - val_loss: 2.5960 - val_acc: 0.5288
Epoch 89/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2338 - acc: 0.9196 - val_loss: 2.6237 - val_acc: 0.5467
Epoch 90/90
32/32 [==============================] - 0s 4ms/step - loss: 0.2127 - acc: 0.9226 - val_loss: 2.6237 - val_acc: 0.5030
Accuracy is: 50.0
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	0
1	0.57	0.38	0.45	77
2	0.65	0.56	0.60	308
3	0.42	0.57	0.49	167
4	0.23	0.20	0.21	50
5	0.33	0.19	0.24	26
accuracy			0.50	628
macro avg	0.37	0.32	0.33	628
weighted avg	0.53	0.50	0.51	628

Classification report analysis:

We can see that our model is performing relatively better on middle risk class detection.

From the above results, we can see both train accuracy is improving with more epochs, but validation accuracy is jumping around and stays around 53%. The nonincreasing validation accuracy might indicate the model overfitting. Our NN model achieves 0.5 f1-score with 90 epochs.

As we can see that positivity rate works better on multiclass classification.

In our experiments, we found that increasing the epochs would increase train accuracy but not validation accuracy, which indicates the model is overfitting. For example, the below screenshot shows train and validation accuracy with 200 epochs, we can see that the train accuray kept going up to 0.96, but the validation accuracy stays around 0.54. Moreover, the validation f1-score is not getting higher too.

```
Epoch 90/100
32/32 [==============================] - 0s 4ms/step - loss: 0.2234 - acc: 0.9133 - val_loss: 2.5417 - val_acc: 0.5209
Epoch 91/100
32/32 [==============================] - 0s 5ms/step - loss: 0.1784 - acc: 0.9381 - val_loss: 2.5675 - val_acc: 0.5487
Epoch 92/100
32/32 [==============================] - 0s 4ms/step - loss: 0.2662 - acc: 0.9040 - val_loss: 2.5612 - val_acc: 0.5268
Epoch 93/100
32/32 [==============================] - 0s 3ms/step - loss: 0.2263 - acc: 0.9187 - val_loss: 2.5521 - val_acc: 0.5229
Epoch 94/100
32/32 [==============================] - 0s 4ms/step - loss: 0.1482 - acc: 0.9479 - val_loss: 2.5936 - val_acc: 0.5249
Epoch 95/100
32/32 [==============================] - 0s 3ms/step - loss: 0.1442 - acc: 0.9495 - val_loss: 2.6278 - val_acc: 0.5348
Epoch 96/100
32/32 [==============================] - 0s 4ms/step - loss: 0.1160 - acc: 0.9636 - val_loss: 2.6601 - val_acc: 0.5626
Epoch 97/100
32/32 [==============================] - 0s 5ms/step - loss: 0.0904 - acc: 0.9792 - val_loss: 2.8400 - val_acc: 0.5388
Epoch 98/100
32/32 [==============================] - 0s 4ms/step - loss: 0.0750 - acc: 0.9766 - val_loss: 3.0793 - val_acc: 0.5070
Epoch 99/100
32/32 [==============================] - 0s 4ms/step - loss: 0.0772 - acc: 0.9721 - val_loss: 3.2642 - val_acc: 0.5129
Epoch 100/100
32/32 [==============================] - 0s 4ms/step - loss: 0.0948 - acc: 0.9628 - val_loss: 2.9493 - val_acc: 0.5447
Accuracy is: 48.2484076433121
/usr/lib64/python3.6/site-packages/sklearn/metrics/classification.py:1439: UndefinedMetricWarning: Recall and F-score are
th no true samples.
  'recall', 'true', average, warn_for)
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.57      0.40      0.47        68
           2       0.62      0.66      0.64       290
           3       0.42      0.41      0.42       186
           4       0.13      0.16      0.15        62
           5       0.00      0.00      0.00        22

    accuracy                           0.48       628
   macro avg       0.29      0.27      0.28       628
weighted avg       0.49      0.48      0.48       628
```

We added dropout layers to our model, it fixed the overfitting issue, but did not improve the model's performance.

```
# Neural network
model = Sequential()
#model.add(Dense(128,activation='relu'))
#model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.5))
#model.add(Dense(32,activation='relu'))
#model.add(Dense(1, activation='sigmoid'))
#model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.add(Dense(len(class_names),activation='softmax'))
model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['acc'])
#model.summary()
history = model.fit(X_train, y_train, validation_split=0.2,epochs=200, batch_size=64)
y_pred = model.predict(X_test)
```

```
32/32 [==============================] - 0s 4ms/step - loss: 0.9286 - acc: 0.6081 - val_loss: 1.0488 - val_acc: 0.5586
Epoch 195/200
32/32 [==============================] - 0s 4ms/step - loss: 0.9306 - acc: 0.6076 - val_loss: 1.0562 - val_acc: 0.5487
Epoch 196/200
32/32 [==============================] - 0s 4ms/step - loss: 0.9466 - acc: 0.5995 - val_loss: 1.0362 - val_acc: 0.5686
Epoch 197/200
32/32 [==============================] - 0s 4ms/step - loss: 0.9202 - acc: 0.5949 - val_loss: 1.0461 - val_acc: 0.5586
Epoch 198/200
32/32 [==============================] - 0s 5ms/step - loss: 0.9469 - acc: 0.5854 - val_loss: 1.0426 - val_acc: 0.5706
Epoch 199/200
32/32 [==============================] - 0s 5ms/step - loss: 0.9586 - acc: 0.5987 - val_loss: 1.0368 - val_acc: 0.5606
Epoch 200/200
32/32 [==============================] - 0s 4ms/step - loss: 0.9882 - acc: 0.5625 - val_loss: 1.0508 - val_acc: 0.5368
Accuracy is: 40.28662420382166
/usr/lib64/python3.6/site-packages/sklearn/metrics/classification.py:1439: UndefinedMetricWarning: Recall and F-score ar
0 in labels with no true samples.
  'recall', 'true', average, warn_for)
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.85      0.33      0.48        69
           2       0.68      0.53      0.60       277
           3       0.50      0.41      0.45       192
           4       0.83      0.08      0.14        65
           5       0.00      0.00      0.00        25

    accuracy                           0.40       628
   macro avg       0.48      0.22      0.28       628
weighted avg       0.64      0.40      0.47       628
```

In our experiment, we explored whether increasing the hidden layers would increase the f1-score significantly. Moreover, we explored increasing the number of nodes in each hidden layer. However, our model didn't get significant improvement with increasing the number of hidden layers or hidden nodes. The reason could be the datasets are not big enough, and the way we class the label into six different classes is too detailed. We could increase the dataset size and modify the six different classes into three or four classes to test again the NN model in the future.

Results Summary:
Recall and F1-Score were given greater attention as performance measures because it was deemed worse for a model to have false negatives than false positives (be too cautious rather than not cautious enough) when determining COVID risk.

*highest percentage is highlighted below

Positivity Rate Test Accuracy (2-Class)				
Model	**Train Accuracy**	**Test Accuracy**	**Recall for Class 1**	**F1-Score**
Logistic Regression	70.10%	70.38%	67.00%	70.00%
SVM	70.74%	70.70%	74.00%	71.00%
Neural Network	75.00%	76.00%	76.00%	76.00%

Fatality Rate Test Accuracy (2-Class)				
Model	**Train Accuracy**	**Test Accuracy**	**Recall for Class 1**	**F1-Score**
Logistic Regression	68.65%	71.34%	71.00%	71.00%
SVM	68.87%	69.11%	55.00%	60.00%
Neural Network	72.00%	73.00%	59.00%	73.00%

Positivity Rate Test Accuracy (6-Class)				
Model	**Train Accuracy**	**Test Accuracy**	**Recall**	**F1-Score**
Logistic Regression	51.51%	49.52%	50.00%	50.00%
SVM	50.84%	49.20%	49.00%	43.00%
Neural Network	52.00%	50.00%	54.00%	50.00%

Fatality Rate Test Accuracy (6-Class)				
Model	**Train Accuracy**	**Test Accuracy**	**Recall**	**F1-Score**
Logistic Regression	51.99%	51.43%	51.00%	51.00%
SVM	51.83%	51.43%	51.00%	43.00%
Neural Network	54.00%	53.00%	50.00%	53.00%

All three models perform very similarly across the board. The neural network performed the best or tied for best in 14 of 16 parameters in the summary.

For 2-class positivity rate, the models are performed similarly for recall and F1-score as well, with the neural network performing slightly better than the other two models. However, the performance gap is not very large.

Model performance for positivity rate and fatality rate were close for 2-class and 6-class, with 2-class values around 0.7 and 6-class values around 0.5. The models did not seem to have more difficulty predicting positivity rate than fatality rate or vice versa.

Conclusion

We can see here that the neural network model generally outperforms the other models, though the difference in performance is not large.

Occupations in production, transportation, and material moving were associated with higher positivity rates, indicating that people who travel for their job are more likely to spread COVID.

Age and diabetes are risk factors for fatality rate, but not positivity rate. This indicates that older, more sickly people are less likely to spread COVID , but more likely to die from COVID if they catch it.

Education level appears to be connected with COVID spread as several features with coefficients of greater magnitude in logistic regression were related to education (e.g. School Enrollment in Fatality, At least Bachelor's Degree, Less Than High School Diploma and Graduate Degree in Positivity).

The Gini Coefficient was also correlated with probability of being high risk for both positivity rate and fatality rate in binary classification, indicating that counties with greater income inequality are hit harder by COVID.

The latitude and longitude of a county also were strongly relevant features for binary fatality rate classification, indicating that certain geographical regions are hit harder by COVID deaths. This makes intuitive sense, as a COVID breakout in one county would be likely to affect neighboring counties.

Potential future steps would include redefining the multiclass in terms of percentiles instead of average and standard deviation. Defining classes in terms of percentiles would lead to more balanced classes which may improve model performance for multi-class classification.

Furthermore, we could explore more classification algorithms to find a better-fitting model.