Hrishikesh Bagalkote

Science Fair: Data Analysis and CERA

At this point, your research should be complete. Submit the information listed below. A scoring rubric is provided separately. Delete the instructional contents within each orange box as you fill it in with your information.

I. Hypothesis (Engineering)

A. Null Hypothesis

Epochs vs. Runtime Null Hypothesis:
The training runtime will increase in a linear manner relative to epochs and will not plateau off.

High vs. Low Positive Factor Models Null Hypothesis:
The predicted relative percent risk of randomly selected data points will not be statistically significant when predicted on high positive factor models (Top 20 Deathcount, Top 20 Casecount, Top 20 Population Density) versus low positive factor models (Lowest 20 Deathcount, Lowest 20 Casecount, Lowest 20 Population Density).

B. Alternative Hypothesis

Epochs vs. Training Runtime Alternate Hypothesis:
The training runtime will increase relative to epochs in a logarithmic manner relative to epochs and will plateau off.

High vs. Low Positive Factor Models Alternate Hypothesis:
The predicted relative risk of randomly selected data points will be statistically significant when predicted on high positive factor models (Top 20 Deathcount, Top 20 Casecount, Top 20 Population Density) versus low positive factor models (Lowest 20 Deathcount, Lowest 20 Casecount, Lowest 20 Population Density).

II. Data Table(s)

Hrishikesh Bagalkote

Epochs (Training Cycles) vs. Average Training Runtimes (ms)

Epochs	Average Training Runtime (ms)
1000	0.998
2000	2.991
3000	3.952
4000	5.022
5000	8.119
6000	9.122
7000	9.671
8000	9.884
9000	9.959
10000	10.187
100000	26.34
200000	30.552
300000	34.102
400000	37.981
500000	40.649
600000	44.101
700000	46.334
800000	47.812
900000	48.445
1000000	49.918

The above table is processed data. Average training runtime values are average trial results, rather than singular trial points. The goal of this portion of data collection was to find the relationship between the number of epochs (training cycles) and the time it takes to complete training.

Average Predicted Percent Risk per Trial in Highest and Lowest 20 Deathcount Models:

Hrishikesh Bagalkote

Trial	Highest 20 Deathcount Average Percent Risk in Trial	Lowest 20 Deathcount Average Percent Risk in Trial
1	50.8142238	153.2567767
2	57.1299558	126.5063073
3	37.61708443	89.47228216
4	45.87349575	63.83847371
5	35.2692335	98.54956186
6	41.96393818	60.99815868
7	48.3331354	121.2244072
8	45.65122952	74.26598997
9	45.84826521	131.2229748
10	30.78568241	57.58090632
Mean of Means	**43.9286244**	**97.69158387**
TTEST VALUE	6.38E-04	

The above table is processed data. Each trial is an average of five randomly picked data points from the holistic dataset. The same five points are forecasted using both models in each respective trial, yielding the average predicted percent risk. The purpose of this portion of the data collection was to determine statistical significance between a model trained on the data points with the highest 20 deathcounts and a model trained on the data points with the lowest 20 deathcounts.

Average Predicted Percent Risk per Trial in Highest and Lowest 20 Casecount Models:

Trial	Highest 20 Casecount Average Percent Risk in Trial	Lowest 20 Casecount Average Percent Risk in Trial
1	50.9505758	162.154531
2	66.1350466	155.0401611
3	40.33867067	103.2852094
4	43.11123286	81.64784681
5	53.48166754	120.1175272
6	47.32847571	85.70105183
7	51.51106743	132.3809187
8	56.5932893	109.3522194
9	41.35799654	157.5939969
10	35.20407815	67.7042858
Mean of Means	**48.60121006**	117.4977748
	TTEST VALUE	8.68E-05

The above table is processed data. Each trial is an average of five randomly picked data points from the holistic dataset. The same five points are forecasted using both models in each respective trial, yielding the average predicted percent risk. The purpose of this portion of the data collection was to determine statistical significance between a model trained on the data points with the highest 20 casecounts and a model trained on the data points with the lowest 20 casecounts.

Average Predicted Percent Risk per Trial in Highest and Lowest 20 Population Density Models:

Hrishikesh Bagalkote

Trial	Highest 20 Population Density Average Percent Risk in Trial	Lowest 20 Population Density Average Percent Risk in Trial
1	52.07231586	219.4504697
2	53.59747298	488.399429
3	36.38223421	227.576314
4	36.05951393	254.5348118
5	44.84551887	351.3613694
6	35.50982698	359.233512
7	46.68497236	294.3919532
8	44.41208336	406.4147313
9	41.78861631	205.0260956
10	31.88354276	156.2678636
Mean of Means	**42.32360976**	**296.265655**
	TTEST VALUE	2.71E-05

The above table is processed data. Each trial is an average of five randomly picked data points from the holistic dataset. The same five points are forecasted using both models in each respective trial, yielding the average predicted percent risk. The purpose of this portion of the data collection was to determine statistical significance between a model trained on the data points with the highest 20 population densities and a model trained on the data points with the lowest 20 population densities.

Predicted Relative Risk in the 20 Most Populous GA Counties:

Hrishikesh Bagalkote

County	Predicted Relative Risk (percent)
DeKalb	398.21%
Cobb	311.71%
Clayton	275.09%
Clarke	188.81%
Muscogee	156.46%
Fulton	141.59%
Bibb	121.28%
Forsyth	117.67%
Richmond	115.01%
Gwinnett	110.23%
Henry	108.94%
Cherokee	81.04%
Hall	80.57%
Douglas	77.18%
Paulding	69.32%
Columbia	65.24%
Houston	64.37%
Lowndes	62.80%
Chattam	56.66%
Coweta	50.45%

The above data table is a list of predicted percent risk values for the 20 most populous counties in Georgia. This portion of data collection was performed to see the neural network's risk predictions in smaller geographical areas using the holistic model.

Predicted Relative Risk in All 50 States:

Hrishikesh Bagalkote

State	Relative Risk (Percent)
California	451.7888378
Texas	360.5262575
Florida	324.3895412
New Jersey	263.9288453
New York	255.8585157
Illinois	243.4065493
Massachussetts	197.5516327
Ohio	188.3263431
Pennsylvania	175.5698619
Rhode Island	163.5933116
Georgia	159.8546858
Maryland	149.6300861
Connecticut	148.9003408
Michigan	148.8922408
Tennesee	145.3709563
North Carolina	144.7533839
Indiana	139.3540967
Wisconnsin	133.5124707
Arizona	125.4041942
Virginia	111.1110597
Missouri	110.2620346
Minnesota	109.0892862
Alabama	102.3763398
Louisiana	97.20233178
Delaware	90.18869981
Kentucky	89.22951219
Colorado	87.71144641
South Carolina	86.66466041
Washington	81.92817157
Oklahoma	81.57712672
Iowa	77.30174234
Mississippi	71.05052838
Arkansas	70.48543141
Utah	69.74848865
Kansas	64.656907
Nevada	64.09688913
New Mexico	52.62637893
Nebraska	49.99283908
Oregon	47.83007015
West Virginia	47.80139349
Hawaii	45.562664
Idaho	44.26973119
New Hampshire	42.35284397
South Dakota	36.25840289
Montana	32.81270677
North Dakota	31.84688556
Vermont	26.89867142
Maine	25.81184471
Alaska	22.9038001
Wyoming	21.43372383

Hrishikesh Bagalkote

The above data table is a list of predicted percent risk values for all 50 states in the US. This portion of data collection was to assess the validity of the network's predictions and generate an insight risk map for all US states.

Derived Training Dataset vs. Median Percent Error:

Training Dataset	Median Percent Error (Prediction vs Real)
Complete Dataset	64%
Highest 20 Deathcount	124%
Highest 20 Casecount	152%
Highest 20 Population Density	87%
Lowest 20 Deathcount	66%
Lowest 20 Casecount	56%
Lowest 20 Population Density	41%

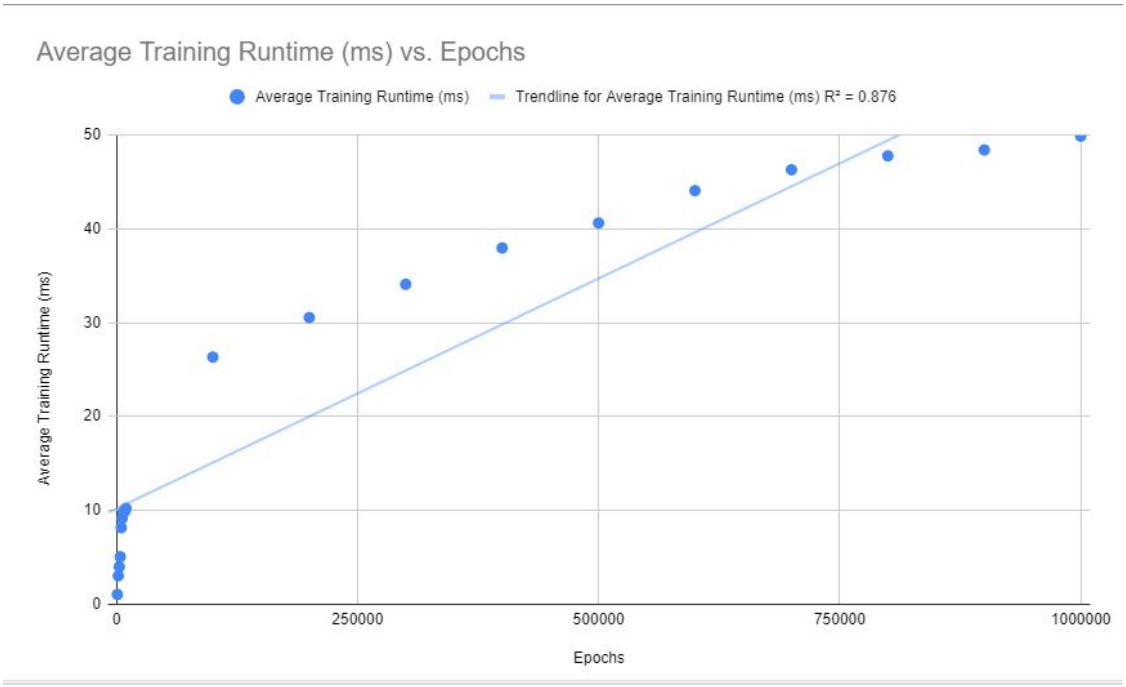
The above data table is processed data obtained from calculating the median percent error between the neural network's predicted deaths and actual deathcount values in the training dataset. In order to do this, the neural network was reprogrammed to display a forecasted deathcount instead of a percent error. The reason the percent error values are so high is because the network was geared to forecast percent risk, not deathcount of a geographical area. However, it can be seen that the lowest 20 casecount and population density models are correlated with having the highest accuracy. Higher percent error values between real and predicted deathcounts are not an issue, as they do not impact percent risk predictions.

III. Analysis

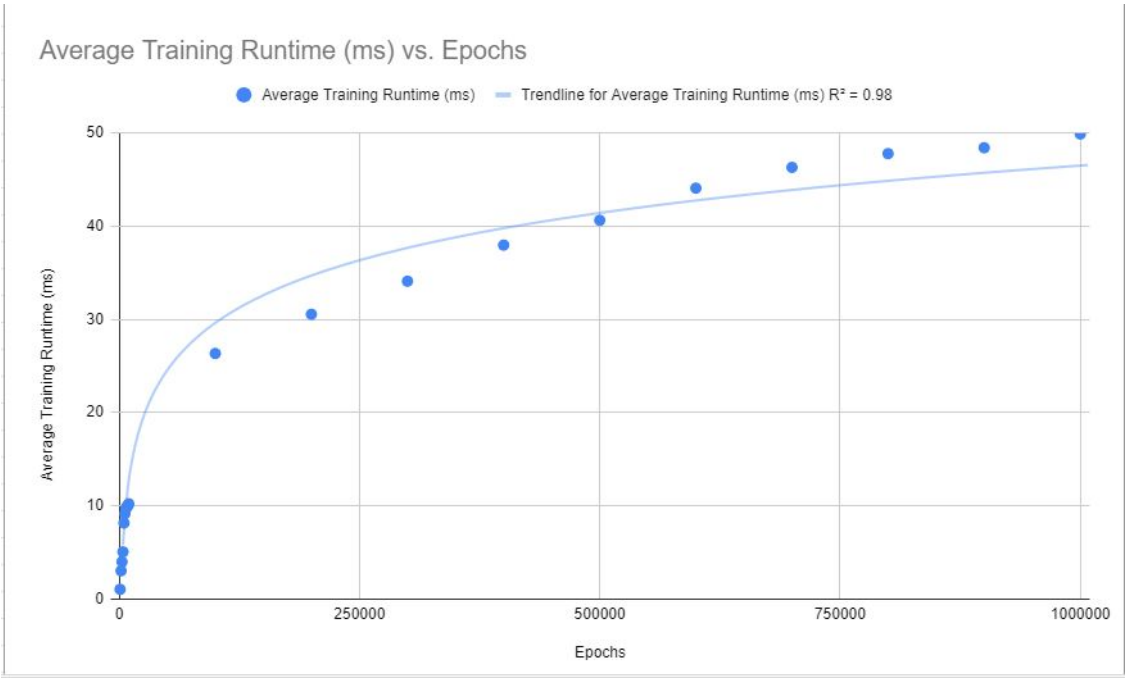
A. Graphical analysis

Epochs (Training Cycles) vs. Average Training Runtime (ms)

Hrishikesh Bagalkote



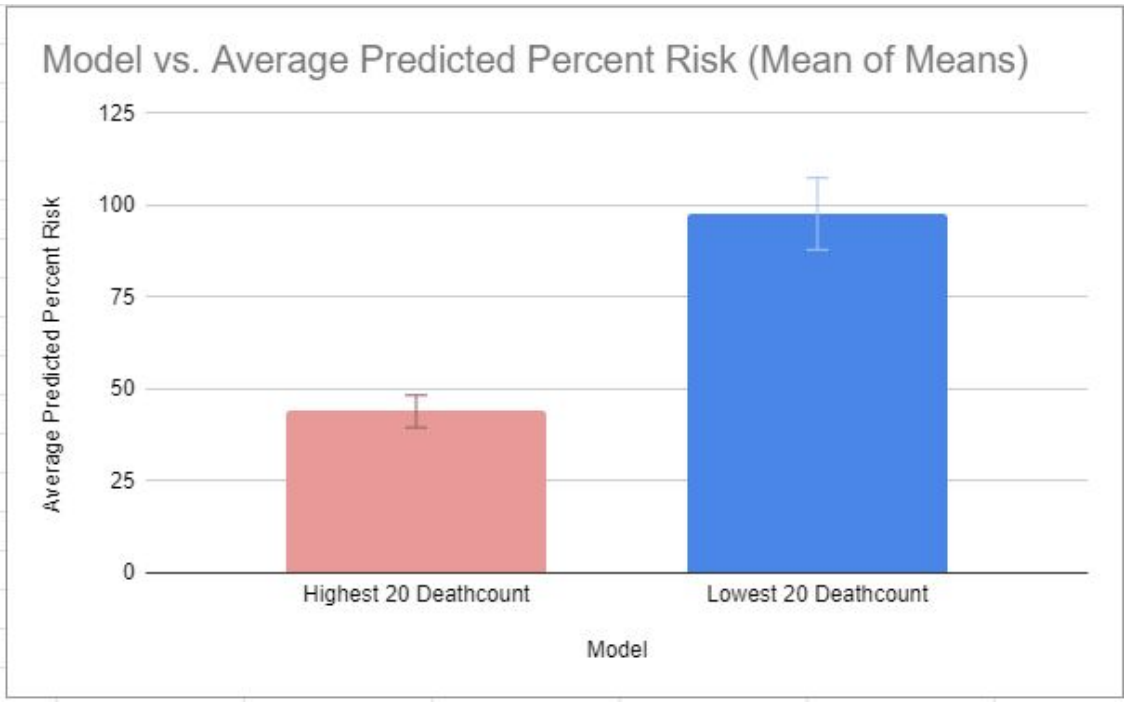
Although linear regression can be shown, a line of best fit does not best represent the data shown in the graph.



A logarithmic function of best fit more accurately fits the data shown in the graph. Training runtimes level off over time as the network completes more epochs.The

logarithmic function seems more representative of the reduction in learning rate in the network as more epochs are completed.

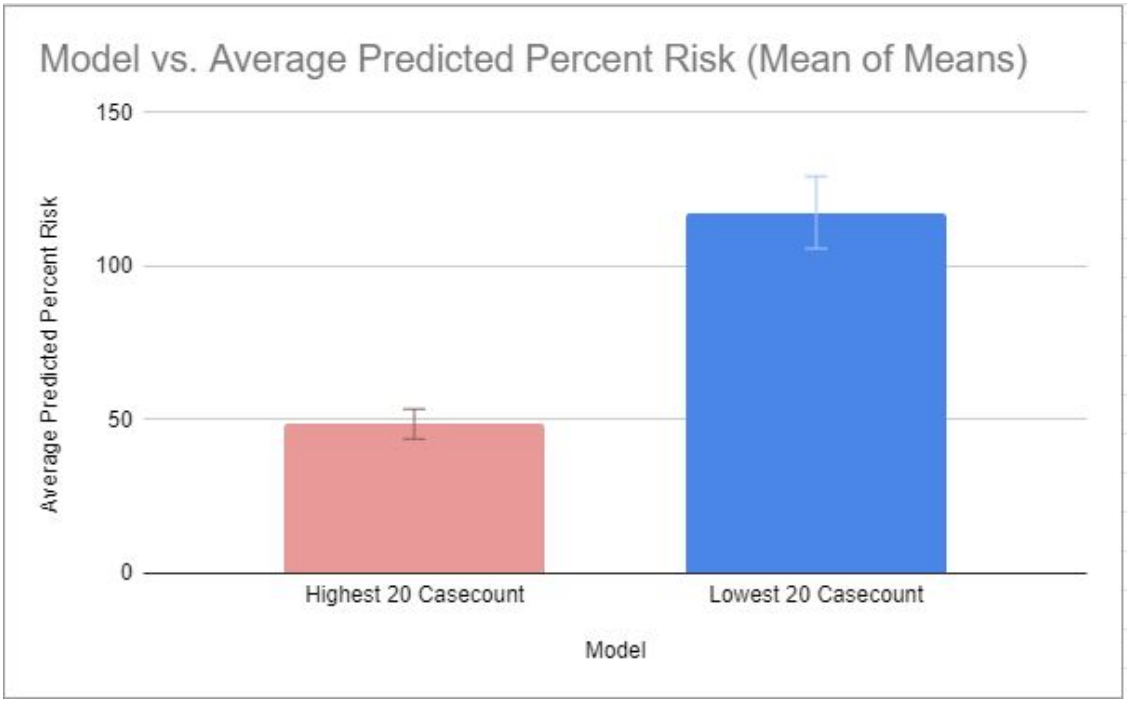
Average Predicted Percent Risk per Trial in Highest and Lowest 20 Deathcount Models:



The lowest 20 deathcount model produces significantly higher risk percentages than the highest 20 deathcount model.

Average Predicted Percent Risk per Trial in Highest and Lowest 20 Casecount Models:

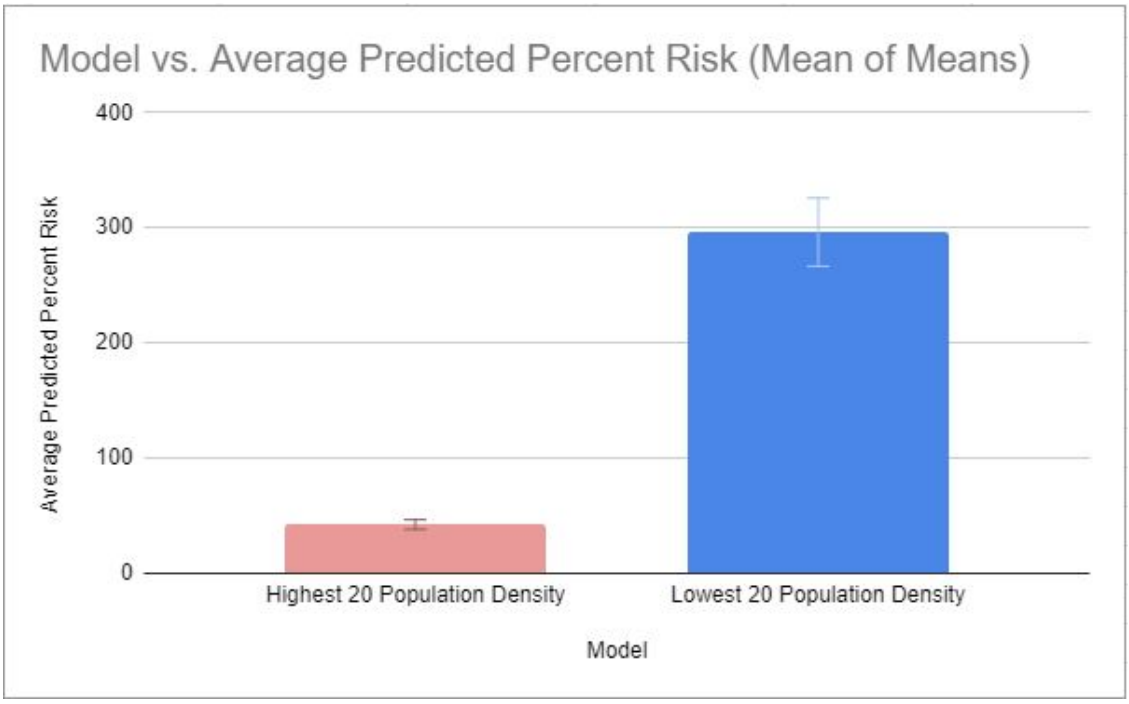
Hrishikesh Bagalkote



The lowest 20 casecount model produces significantly higher risk percentages than the highest 20 casecount model.

Average Predicted Percent Risk per Trial in Highest and Lowest 20 Population Density Models:

Hrishikesh Bagalkote



The lowest 20 population density model produces significantly higher risk percentages than the highest 20 casecount model.

B. Statistical analysis

Linear vs. Logarithmic Line of Best Fit:

Linear (equation: 4.91e-5x + 10.2)			
Test (Epochs in Millions)	Average Real Runtime (ms)	Modeled Runtime Through Equation(ms)	Percent Error
10	305.117	501.2	39.12270551
11	354.915	550.3	35.50517899
12	361.026	599.4	39.76876877
13	376.192	648.3	41.97254358
14	402.389	697.6	42.3180906
15	445.002	746.7	40.40417838
Average Percent Error:	39.84857764		
T Test (p) Value:	0.0004575532681		
Reject Null Hypothesis?	No		
df:	5		
Critical Value (p cutoff):	0.01		

Hrishikesh Bagalkote

Logarithmic (equation: -54.9 + 7.34lnx)				
Test (Epochs in Millions)	Average Real Runtime (ms)	Modeled Runtime Through Equation(ms)	Percent Error	
10	305.117	63.407	381.203968	
11	354.915	64.106	453.637725	
12	361.026	64.745	457.6121708	
13	376.192	65.333	475.8070194	
14	402.389	65.877	510.8186469	
15	445.002	68.383	550.7494553	
Average Percent Error:	471.6381642			
T Test (p) Value:	0.00001688502351			
Reject Null Hypothesis?	No			
df	5			
Critical Value (p cutoff):	0.01			

The above data was collected for verification purposes of the neural network's optimization. The conclusions show that the neural network's runtimes are shorter than the linear equation's expected values, but longer than the logarithmic equation's expected values. However, the p values and average percent error indicate that the linear equation of best fit is far more statistically similar to real runtime values.

Highest vs. Lowest Deathcount Model Statistics:

Trial	Highest 20 Deathcount Average Percent Risk in Trial	Lowest 20 Deathcount Average Percent Risk in Trial
1	50.8142238	153.2567767
2	57.1299558	126.5063073
3	37.61708443	89.47228216
4	45.87349575	63.83847371
5	35.2692335	98.54956186
6	41.96393818	60.99815868
7	48.3331354	121.2244072
8	45.65122952	74.26598997
9	45.84826521	131.2229748
10	30.78568241	57.58090632
Mean of Means	43.9286244	97.69158387
T Test (p) Value	6.38E-04	
Reject Null Hypothesis?	Yes	
df	9	
Critical Value (p value cutoff)	0.01	

The above data was collected to determine if statistical significance is present between highest and lowest 20 deathcount models. Based on the p value, we can say with over 99% confidence that there is significant difference in the predictions of the two models, as the p value is less than 0.01.

Highest vs. Lowest Casecount Model Statistics:

Hrishikesh Bagalkote

Trial	Highest 20 Casecount Average Percent Risk in Trial	Lowest 20 Casecount Average Percent Risk in Trial
1	50.9505758	162.154531
2	66.1350466	155.0401611
3	40.33867067	103.2852094
4	43.11123286	81.64784681
5	53.48166754	120.1175272
6	47.32847571	85.70105183
7	51.51106743	132.3809187
8	56.5932893	109.3522194
9	41.35799654	157.5939969
10	35.20407815	67.7042858
Mean of Means	**48.60121006**	117.4977748
T Test (p) Value	8.68E-05	
Reject Null Hypothesis?	Yes	
df	9	

The above data was collected to determine if statistical significance is present between highest and lowest 20 casecount models. Based on the p value, we can say with over 99% confidence that there is significant difference in the predictions of the two models, as the p value is less than 0.01.

Highest vs. Lowest Population Density Model Statistics:

Trial	Highest 20 Population Density Average Percent Risk in Tria	Lowest 20 Population Density Average Percent Risk in Trial
1	52.07231586	219.4504697
2	53.59747298	488.399429
3	36.38223421	227.576314
4	36.05951393	254.5348118
5	44.84551887	351.3613694
6	35.50982698	359.233512
7	46.68497236	294.3919532
8	44.41208336	406.4147313
9	41.78861631	205.0260956
10	31.88354276	156.2678636
Mean of Means	**42.32360976**	**296.265655**
T Test (p) Value	2.71E-05	
Reject Null Hypothesis?	Yes	
df	9	
Critical Value (p value cutoff)	0.01	

The above data was collected to determine if statistical significance is present between highest and lowest 20 casecount models. Based on the p value, we can say with over 99% confidence that there is significant difference in the predictions of the two models, as the p value is less than 0.01.

*** The second null hypothesis can be rejected, as all the p values for the model comparison statistical tests were under 0.01, giving a 99% confidence interval for statistical difference. For all comparisons with positive factor linked**

models, the model normalized towards the lower measure consistently predicts higher risk percentages than its respective, opposite, high measure model.

*** The above "trial" values in the tables are averages of 5 randomly selected data points forecasted on both models. This totals to 10 groups of 5 randomly selected input points forecasted on both models to determine statistical significance.**

IV. Conclusion- CERA (Claim, Evidence, Reasoning, Application)

A. Claim

Epochs vs. Runtime:
The linear equation derived from regression represents the relationship between epochs and runtime. As epochs increase, training runtime increases at a generally linear rate. (Accept null hypothesis).

High vs. Low Positive Factor Models:
The predicted percent risk when forecasted from a high positive factor model is statistically significant and different from the predicted percent risk when forecasted from a low positive factor model. (Reject null hypothesis).

B. Evidence

Methodology:
We used a combination of graph analysis and T test (p value) statistical measures to gauge statistical significance/difference.

Epochs vs. Runtime Statistical Measures:
By looking at the line graph of epochs (x-axis) vs. average training runtime (y-axis), we saw that the logarithmic equation of best fit conformed to the data points in the graph more tightly than the linear equation of best fit. Although the logarithmic function seemed to be more representative of the window of data our graph displayed, the statistical tests and measures calculated suggested otherwise. The T test value for the linear equation was greater than the T test value for the logarithmic equation. This means that the training runtimes forecasted by the linear equation were closer to the real values than that of the logarithmic equation. More importantly, we calculated percent error for each prediction compared to the actual training runtime value in milliseconds. The linear function had a far lower average percent error than the logarithmic function, telling us that the training runtimes will increase at a generally

linear rate as epochs increase proportionally. Due to the linear equation of best fit being more representative of the data, we accepted the first null hypothesis.

High vs. Low Positive Factor Model Statistical Measures:
By looking at the average predicted percent risk values, it can be seen that the low positive factor models predicted higher risk percentages than the high positive factore models did for the same randomly selected data points.

- The lowest 20 deathcount model predicted around a 54% higher average risk than the top 20 deathcount model.
- The lowest 20 casecount model predicted around a 69% higher average risk than the top 20 casecount model.
- The lowest 20 population density model predicted around a 254% higher average risk than the top 20 population density model.

To determine if these disparities were really statistically significant, or simply due to chance, we performed another T test. In all three comparisons, the resulting value from the T test (p value) was a number under 0.01 which was extremely close to zero. This allowed us to confirm that the differences in model predictions were indeed statistically significant, and to reject the second null hypothesis.

C. Reasoning

Statistical Reasoning:
Epochs vs. Runtime: As stated before, the linear equation of best fit better represented the relationship between epochs and runtime than the logarithmic equation of best fit. Since the linear equation of best fit had a higher p value when being compared to real runtime values, it was less statistically different and varied than the logarithmic equation of best fit. In addition to this, percent error between expected and real runtime values was far lower in the linear equation of best fit than the logarithmic equation of best fit. The linear equation always modeled a higher training runtime than the real runtime value at a certain epoch x-value, but had lower error in comparison to the logarithmic function at higher epoch values.

High vs. Low Positive Factor Models: Similarly to the statistical tests performed in the epochs vs. runtime data, T test values indicated that high and low positive factor models were extremely statistically significant (with p values less than 0.01). The high disparity in average percentage risks predicted show that low positive factor models consistently forecast much higher risk percentages for the same input data than high positive factor models.

Hrishikesh Bagalkote

Logical Reasoning (technicalities in the network, algorithms, and codebase):
Epochs vs. Runtime: When writing high performance algorithms, time complexity is a major factor a programmer must consider. In summary, time complexity is the amount of time in arbitrary units to a scale an algorithm takes to complete based on the operations that take place within the algorithm. In neural networks and machine learning, the neuron does not learn at the same rate as epochs pass. This means that the algorithm slowly does less and less math in each training cycle. Our neural network specifically adjusts weights corresponding to each input feature according to the error of the last epoch's prediction. The more epochs pass, the less the error in the network's predictions during training. Due to these factors, the learning rate of our neural network decreases over time as epochs pass. When the learning rate of the algorithm slows down, the amount of time spent per epoch decreases as well. Because of this, training runtime does not increase at a set linear rate as epochs pass. However, the linear equation of best fit was still more accurate in extrapolating our runtime data in comparison to the logarithmic equation of best fit solely because the logarithmic function plateaued off much more sharply than our runtimes did. In a perfect scenario, we would be able to measure runtime down to the nanosecond and observe runtime per epoch decreasing over millions of epochs.

High vs. Low Positive Factor Models: The way our neural network creates models is through weight correction over time and normalization around a baseline. When a network is trained on a low positive factor dataset, such as the lowest 20 deathcount data points, the model is normalized around a far lower relative risk. Whereas when a network is trained on a high positive factor dataset, such as the top 20 population density data points, the model's baseline risk is significantly higher. When a model's relative average risk perception is low, its risk predictions on new data points will be high, as the new data points far surpass the model's risk baseline. On the other hand, a model's risk predictions will be low for new data points if its existing risk baseline is significantly higher than that of the new data points.

Different Approaches:
In retrospect, our group might have collected more data on epochs and their impact on training runtimes. Creating a graph that encompasses a greater window of data would allow us to calculate a logarithmic function of best fit that was more representative than the linear equation.
Another area we can focus on in the future is the development of a GUI application that shows the impact of each input feature in the model during the prediction sequence

of the neural network. Doing this would increase the usability of our algorithm and provide more insight on the model's reasoning behind its predictions.

D. Application

The main focus of this project was to build an efficient neural network that can predict relative risk of specific areas with ease. Traditional mathematical forecasting models are used to forecast risk in larger geographical areas, but our neural network allows researchers and trend forecasters to predict risk profiles of smaller areas in a simple manner. Using our software and algorithm, the relative risk of areas such as school districts, counties, and parks can be forecasted. The neuron is extendible, meaning that it will train on as many input features given to produce a model with no code modifications. One can train on additional features including but not limited to social distancing levels, hospitalization rates, demographic factors, and preexisting conditions. Now that we have a neuron and engine that is optimized to calculate relative risk, there are a lot of things we can do with our codebase and data.

One viable idea is to create another neuron alongside the existing one that is optimized to forecast deathcount of an area. Writing a new algorithm and tweaking it so that deaths can be forecasted with little error is a path we could take to extend our research.

Another continuation of our research we could pursue is the construction of a website or application that forecasts COVID-19 deathcounts and relative risk of areas all over the world over time. By utilizing an API to feed new data into our dataset and programming a backend software to periodically generate new models, our insights would be accessible to a larger number of people.

Finally, we can take our existing optimized neural network and test it against other use cases. By modifying our existing algorithms for classification problems or forecasting tasks unrelated to COVID-19, we can make lightweight machine learning available to everyone without the need for high performance GPU arrays or servers.