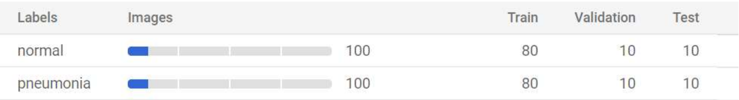
AutoML Modeling Report

Amisha Muni

Binary Classifier with Clean/Balanced Data

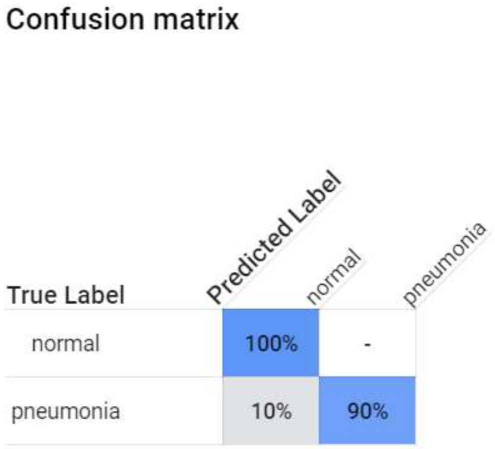
Train/Test Split
How much data was used for training? How much data was used for testing?

100 normal and 100 pneumonia images were uploaded. From these 80 of each type were used as training data, 10 of each type for validation, and 10 of each type for test data – i.e. 80% of each type as training, 10% of each type as validation and 10% of each type as test.



Labels	Images		Train	Validation	Test
normal		100	80	10	10
pneumonia		100	80	10	10

Confusion Matrix
What do each of the cells in the confusion matrix describe? What values did you observe (include a screenshot)? What is the true positive rate for the “pneumonia” class? What is the false positive rate for the “normal” class?

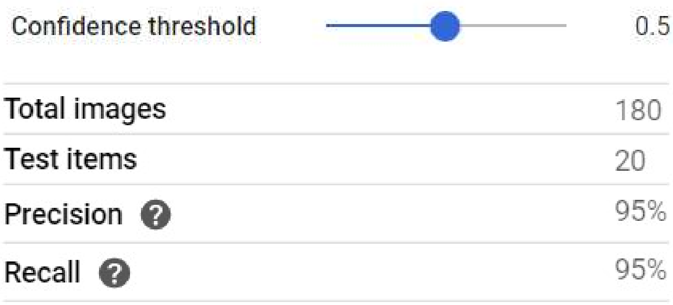


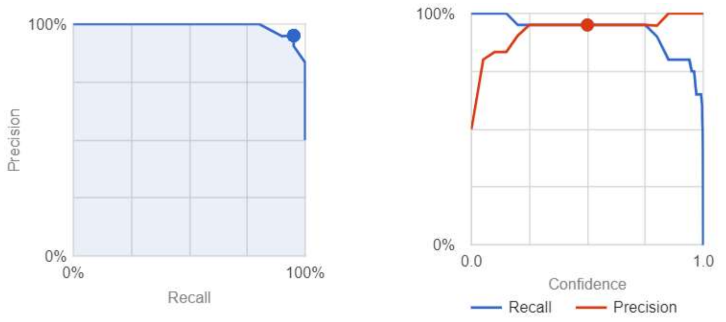
Confusion matrix

True Label	Predicted Label normal	pneumonia
normal	100%	-
pneumonia	10%	90%

The Confusion matrix is a grid showing all the Predicted labels against the True labels, and its aimed at giving us some insight into where we should improve our training data to increase the models accuracy.

Cell: True label normal, Predicted label normal: This tells us all test data which was predicted to be normal, was actually normal (100%)

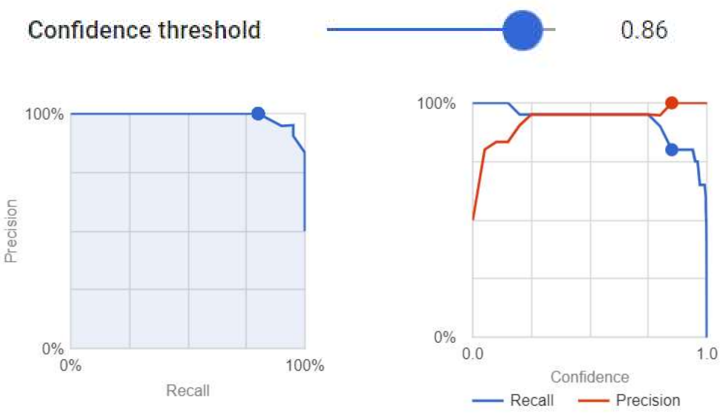
	Cell: True label normal, Predicted label pneumonia: This tells us there was no test data which was predicted to be pneumonia, which was actually normal **Cell: True label pneumonia, Predicted label normal:** This tells us that 10% of test data was predicted as normal, though it was really pneumonia. i.e. the model has some errors and is considering some pneumonia cases as normal. **Cell: True label pneumonia, Predicted label pneumonia:** This tells us that 90% of test data was predicted correctly as pneumonia, when it was really pneumonia. True positive for Pneumonia class: 90% False positive for Normal class: 10%
Precision and Recall What does precision measure? What does recall measure? What precision and recall did the model achieve (report the values for a score threshold of 0.5)?	Precision measures how likely a prediction by a model is to be correct. – i.e. true predictions by total predictions. Recall indicates how good the model is at identifying actual occurrences – i.e. percentage of correctly identified instances vs total possible instances Model achieved a Precision and Recall of 95%, with a threshold of 0.5  Confidence threshold 0.5 Total images 180 Test items 20 Precision 95% Recall 95%

<table>
<tr>
<td></td>
<td>


</td>
</tr>
<tr>
<td>
<b>Score Threshold</b>
When you increase the threshold what happens to precision? What happens to recall? Why?
</td>
<td>
This model has a long threshold range over which the Precision and Recall are steady and at 95% (refer the graph)

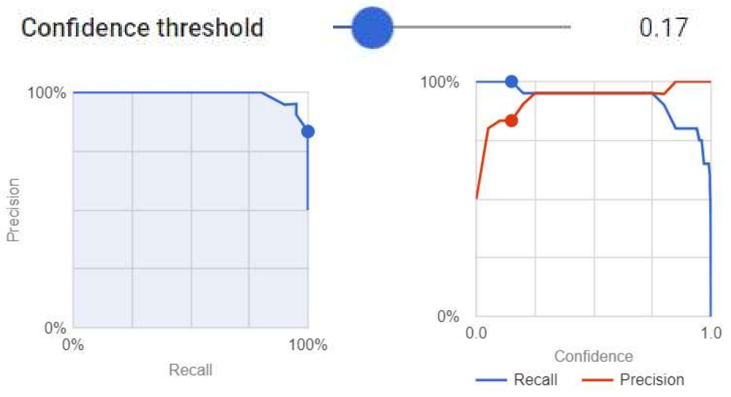
On increasing the threshold past this range, the Precision increases, and the Recall decreases.

e.g. with threshold value 0.86



Confidence threshold 0.86

Similarly, on reducing the threshold below this range, the Precision decreases and the Recall increases
e.g. with threshold value 0.17

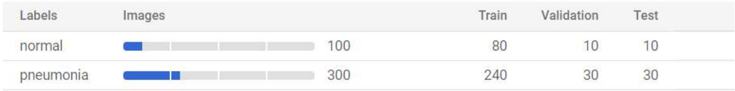


Confidence threshold 0.17

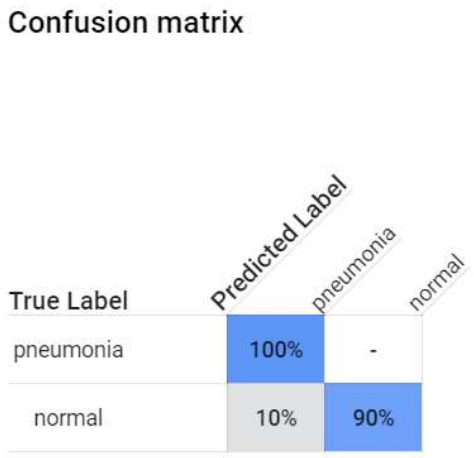
</td>
</tr>
</table>

	Why happens to Precision and Recall with Threshold value increase: The threshold is the threshold of an Activation function, which decides whether to classify an image as normal or pneumonia. When the threshold is low, we are less Precise. When the threshold is high, we are more Precise, essentially we have stricter criteria for classifying the image. This leads to Less false positives, and therefore Precision is higher with a high threshold. Recall on the other hand is the ratio of true positives to the sum of true positives and false negatives. With a higher threshold, we have stricter criteria and are likely to have more false negatives. This leads to a lower Recall value.

Binary Classifier with Clean/Unbalanced Data

<table>
<tr>
<td><b>Train/Test Split</b>
How much data was used for training? How much data was used for testing?</td>
<td>100 normal and 300 pneumonia images were uploaded. From these 10% of each type were used for validation and 10% of each type for test, leaving 80% of each type as training data.

<table>
<tr><th>Labels</th><th>Images</th><th>Train</th><th>Validation</th><th>Test</th></tr>
<tr><td>normal</td><td>100</td><td>80</td><td>10</td><td>10</td></tr>
<tr><td>pneumonia</td><td>300</td><td>240</td><td>30</td><td>30</td></tr>
</table>
</td>
</tr>
</table>

Confusion Matrix
How has the confusion matrix been affected by the unbalanced data? Include a screenshot of the new confusion matrix.



Confusion matrix

	Predicted Label	
True Label	pneumonia	normal
pneumonia	100%	-
normal	10%	90%

The impact to the confusion matrix is pretty much the exact opposite of the case with balanced data in this case.

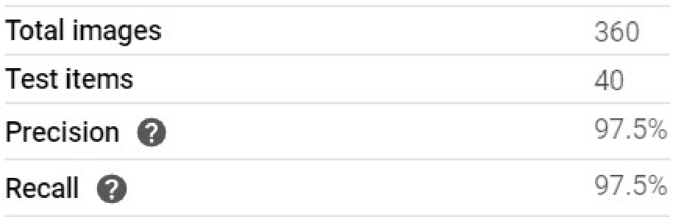
The model shows a skew towards labelling normal cases as pneumonia, reflected by the values shown – 10% of true label normal data has been predicted by the model to be pneumonia, and it's a false positive prediction.

Precision and Recall
How have the model's precision and recall been affected by the unbalanced data (report the values for a score threshold of 0.5)?

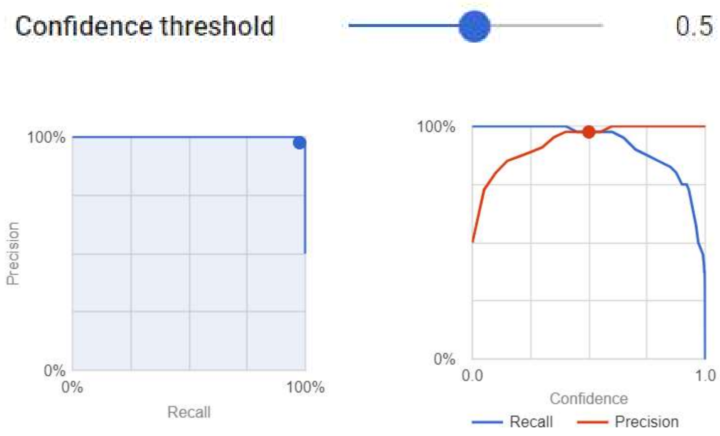
Precision and Recall have both increased to 97.5% from the earlier 95% in the first exercise.

Noted I was actually expecting the Precision to go down and Recall to go up, but its hard to conclusively know why the Precision has gone up here – it would depend really on the actual content of the image files I landed up choosing to train my model.

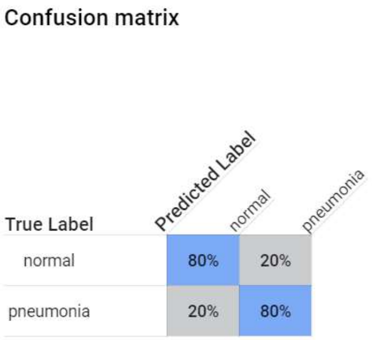
Perhaps a bigger sample size than that suggested for this exercise will show this more clearly. Choosing a different threshold value or more test cases may also give a better indication of actually how well the model is performing.

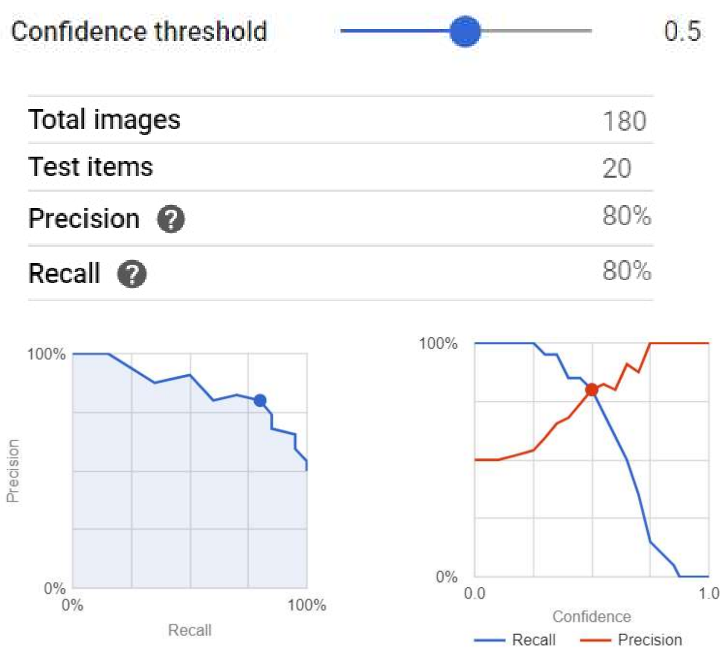


Total images	360
Test items	40
Precision ?	97.5%
Recall ?	97.5%

	 Comparing with the earlier graphs, there is a narrower range of the threshold value for which the Precision and Recall stay at this value.
Unbalanced Classes From what you have observed, how do unbalanced classed affect a machine learning model?	Unbalanced classes add a skew towards a Prediction for the class which has more samples. When a model trained on Unbalanced data is subjected to real data which it has to classify – we can have more false positive predictions for the class which had more training data, and incorrect classification.

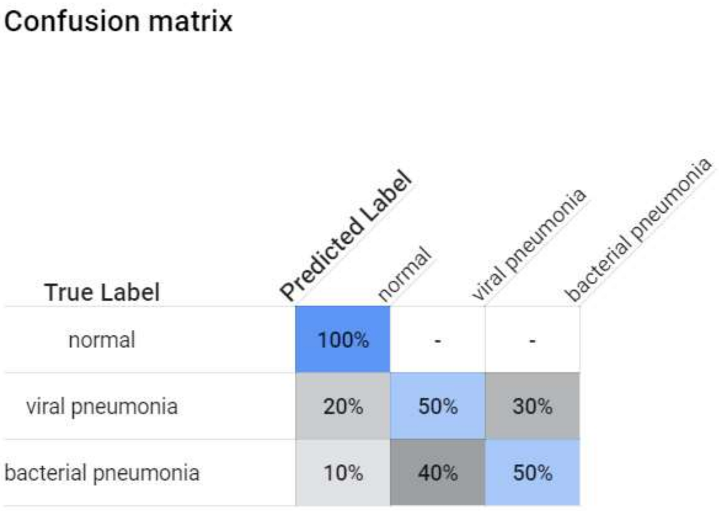
Binary Classifier with Dirty/Balanced Data

<table>
<tr>
<td><b>Confusion Matrix</b>
How has the confusion matrix been affected by the dirty data? Include a screenshot of the new confusion matrix.</td>
<td>
<b>Confusion matrix</b>
<table>
<tr><th>True Label</th><th>Predicted Label
normal</th><th>pneumonia</th></tr>
<tr><td>normal</td><td>80%</td><td>20%</td></tr>
<tr><td>pneumonia</td><td>20%</td><td>80%</td></tr>
</table>
Seeing the confusion matrix, it is clear that we have more errors in the classification.
20% of actual normal cases are inaccurately predicted as pneumonia, and 20% of actual pneumonia cases are</td>
</tr>
</table>

	inaccurately predicted as normal.
Precision and Recall How have the model's precision and recall been affected by the dirty data (report the values for a score threshold of 0.5)? Of the binary classifiers, which has the highest precision? Which has the highest recall?	 Confidence threshold 0.5 Total images 180 Test items 20 Precision 80% Recall 80% 100% Precision 0% 0% 100% Recall 100% 0% 0.0 1.0 Confidence Recall Precision Both Precision and Recall have reduced from the previous iterations. As the data that the model is trained on is dirty, the model now has less accuracy in the predictions it makes. Of the 3 attempts around Binary classifiers in this project, the highest Precision and Recall both have been for the unbalanced dataset (however please note my comments above, as it should really have had a lower Precision, and the first model should have had the best)
Dirty Data From what you have observed, how does dirty data affect a machine learning model?	Dirty data reduces the Precision as well as the Recall, as it reduces the correct predictions made by the model. There is a degradation of the performance of the model.

3-Class Model

Confusion Matrix
Summarize the 3-class confusion matrix. Which classes is the model most likely to confuse? Which class(es) is the model most likely to get right? Why might you do to try to remedy the model's "confusion"? Include a screenshot of the new confusion matrix.



Confusion matrix

Predicted Label

True Label	normal	viral pneumonia	bacterial pneumonia
normal	100%	-	-
viral pneumonia	20%	50%	30%
bacterial pneumonia	10%	40%	50%

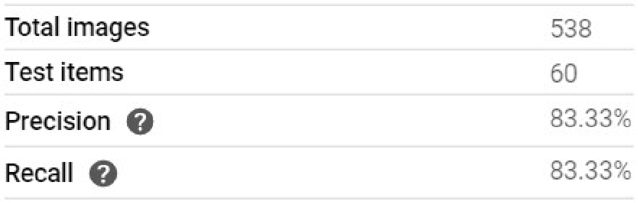
The model is most likely to get the normal class right – as can be seen from the confusion matrix. Every case which is really normal has been predicted as normal. However 20% of viral pneumonia cases and 10% of bacterial pneumonia cases have been incorrectly classified as normal.

The model is most likely to confuse the bacterial and viral pneumonia classes. As can be seen from the confusion matrix, 40% of bacterial pneumonia cases have been incorrectly predicted as viral, and 30% of viral pneumonia cases have been incorrectly predicted as bacterial.

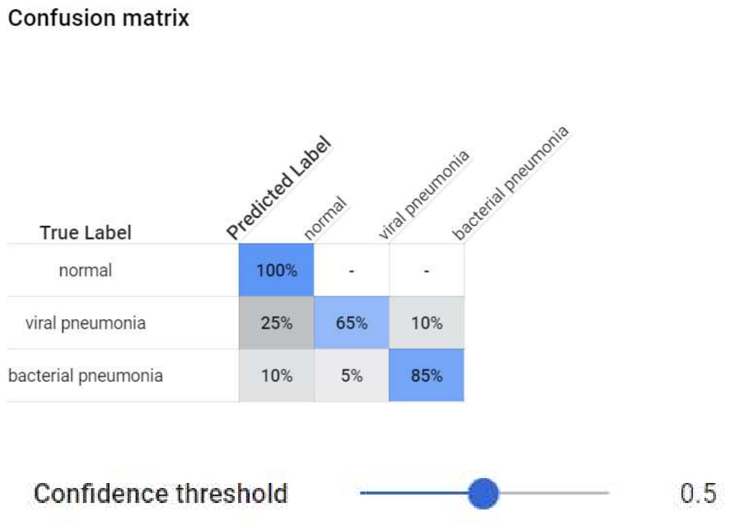
To try to remedy the model's confusion, we can:

- Increase the training data uniformly
- Increase the threshold (this can increase the Precision)

Refer here the result after increasing the training dataset (100 more images added for normal, 100 more for bacterial and 98 more for viral)



Total images	538
Test items	60
Precision ?	83.33%
Recall ?	83.33%

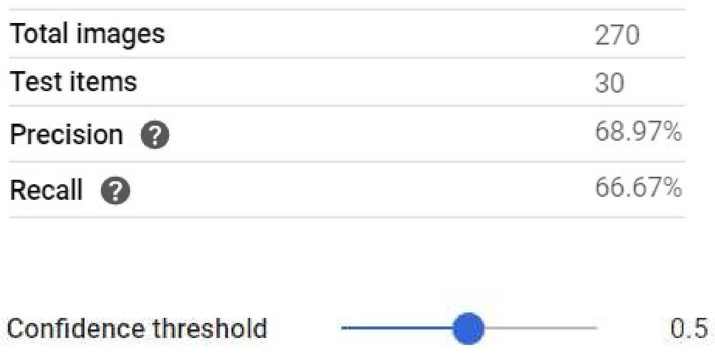


Confusion matrix

True Label	Predicted Label: normal	viral pneumonia	bacterial pneumonia
normal	100%	-	-
viral pneumonia	25%	65%	10%
bacterial pneumonia	10%	5%	85%

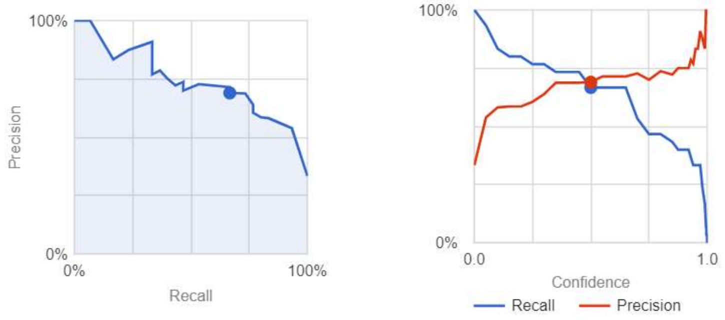
Confidence threshold 0.5

Precision and Recall
What are the model's precision and recall? How are these values calculated (report the values for a score threshold of 0.5)?



Total images	270
Test items	30
Precision ?	68.97%
Recall ?	66.67%

Confidence threshold 0.5



Precision of a class = (Correct predictions for that class) / (Total predictions for that class)

Recall of a class = (Correct predictions for that class) / (Total real instances for that class)

Samples per class = 100

	Pnormal = 100/130 = 0.769
	Pbacterial = 50/80 = 0.625
	Pviral = 50/90 = 0.555
	
	Rnormal = 100/100 = 1
	Rbacterial = 50/100 = 0.5
	Rviral = 50/100 = 0.5
	
	Pmodel = (Pnormal+Pbacterial+Pviral) / 3 = 0.6496 = 64.96%
	Rmodel = (Rnormal+Rbacterial+Rviral) / 3 = 0.6666 = 66.66%

F1 Score What is this model's F1 score?	F1 = (2*Pmodel*Rmodel) / (Pmodel+Rmodel) = 0.6579