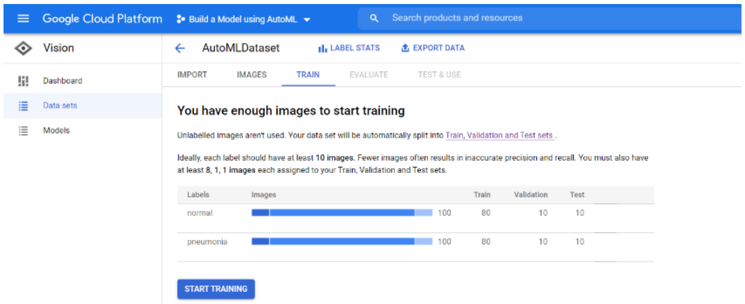
AutoML Modeling Report

<Erkan Hatipoglu >

Binary Classifier with Clean/Balanced 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><i>“<b>Training Dataset:</b> The sample of data used to fit the model.
<b>Validation Dataset:</b> The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.
<b>Test Dataset:</b> The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.” *</i>

There were 200 images in total (100 normal & 100 pneumonia). From this dataset, 160 images (80 normal & 80 pneumonia) were used for training, 20 images were used for testing (10 normal & 10 pneumonia) and 20 images were used for validation (10 normal & 10 pneumonia) as shown in the image below:


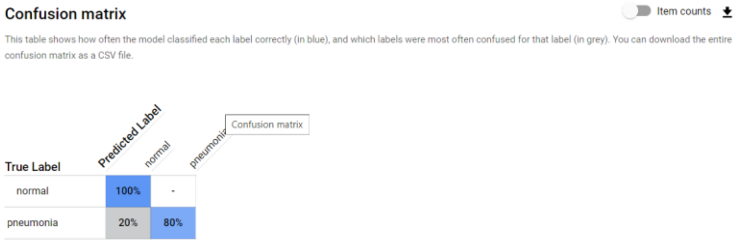
You have enough images to start training
Unlabelled images aren't used. Your data set will be automatically split into Train, Validation and Test sets .
Ideally, each label should have at least 10 images. Fewer images often results in inaccurate precision and recall. You must also have at least 8, 1, 1 images each assigned to your Train, Validation and Test sets.
<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>100</td><td>80</td><td>10</td><td>10</td></tr>
</table>

<b>* https://machinelearningmastery.com/difference-test-validation-datasets/</b></td>
</tr>
<tr>
<td><b>Confusion Matrix</b>
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?</td>
<td>Confusion matrix is a grid which shows all the predicted labels relative to all true labels. In our case we have 4 cells in confusion matrix. For <b><i>normal</i></b> class those cells can be explained as below:
<ol>
<li>True Positives (Row 1 Column 1, TP in short)
Positive labels (actual) that are predicted as positives</li>
<li>False Negatives (Row 1 Column 2, FN in short)
Positive labels (actual) that are predicted as negatives</li>
<li>False Positives (Row 2 Column 1, FP in short)</li>
</ol></td>
</tr>
</table>

Negative labels (actual) that are predicted as positives

4. True Negatives (Row 2 Column 2, TN in short)
   Negative labels (actual) that are predicted as negatives

In Google Vision Confusion matrix can be displayed using percentages or item counts. I have included both images below.



Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey). You can download the entire confusion matrix as a CSV file.

True Label	Predicted Label normal	pneumonia
normal	100%	-
pneumonia	20%	80%

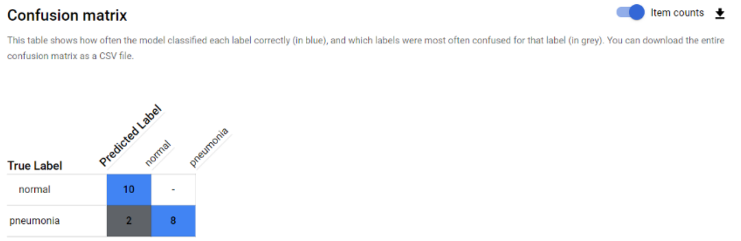
As can be seen from the image below, all the normal images (10 in total) were predicted as normal. On the other hand, 2 of the pneumonia images (10 in total) were predicted as normal and 8 of the pneumonia images were predicted as pneumonia.

As a result, for ***normal*** class:

- TP = 10 (100%),
- FN = 0 (0%),
- FP = 2 (20%),
- TN = 8 (80%).

For ***pneumonia*** class:

- TP = 8 (80%),
- FN = 2 (20%),
- FP = 0 (0%),
- TN = 10 (100%).



Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey). You can download the entire confusion matrix as a CSV file.

True Label	Predicted Label normal	pneumonia
normal	10	-
pneumonia	2	8

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 and recall measure the model such that we can understand how the model performs for an individual class, as well as how it performs across classes.

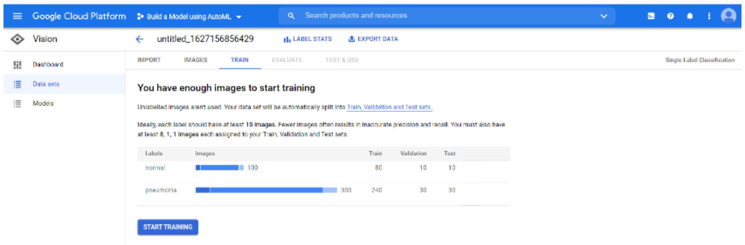
Model precision measures the percentage of correct predictions against total number of predictions.

Model recall measures the percentage of correctly identified instances against total possible instances.

	Normal class ***Precision*** = TP/(TP+FP) = 10/(10+2) = 10/12 ~= 0,83 ***Recall*** = TP/(TP+FN) = 10/(10+0) = 10/10 = 1 ***Pneumonia class*** ***Precision*** = TP/(TP+FP) = 8/(8+0) = 8/8 = 1 ***Recall*** = TP/(TP+FN) = 8/(8+2) = 8/10 = 0,8
Score Threshold When you increase the threshold what happens to precision? What happens to recall? Why?	***Normal class*** When we increase the threshold (0,60 for example) first the number of TP starts to decrease, and the number of FN starts to increase since there are some relatively low-level confidence normal images (P=0,59), but the number of FP stays the same. As a result, both precision and recall decrease. For Confidence threshold = 0,6 ***Precision*** = TP/(TP+FP) = 9/(9+2) = 9/11 ~= 0,81 ***Recall*** = TP/(TP+FN) = 9/(9+1) = 9/10 = 0,9 If we increase the threshold further (0,75 for example) this time both TP and FP decrease and FN increases. The decrease of FP is due to some relatively mid-level confidence pneumonia images (P=0,71). That is some pneumonia images have been given a high probability of being normal. As a result, precision increases but recall decreases. For Confidence threshold = 0,75 ***Precision*** = TP/(TP+FP) = 8/(8+1) = 8/9 ~= 0,89 ***Recall*** = TP/(TP+FN) = 8/(8+2) = 8/10 = 0,8 If we increase the threshold further (0,99 for example) both TP and FP continue to decrease, and TP becomes 1 while FP becomes 0. FN continues to increase and becomes 9. As a result, precision increases to 100% but recall decreases to 10%. For Confidence threshold = 0,99 ***Precision*** = TP/(TP+FP) = 1/(1+0) = 1/1 = 1 ***Recall*** = TP/(TP+FN) = 1/(1+9) = 1/10 = 0,1 If we increase the threshold to 1, both TP and FP continue to decrease. For this case TP and FP becomes 0. FN continues to increase and becomes 10. As a result, precision increases to 100% but recall decreases to 0%. For Confidence threshold = 1 ***Precision*** = TP/(TP+FP) = 0/(0+0) = 0/0 = Unknown

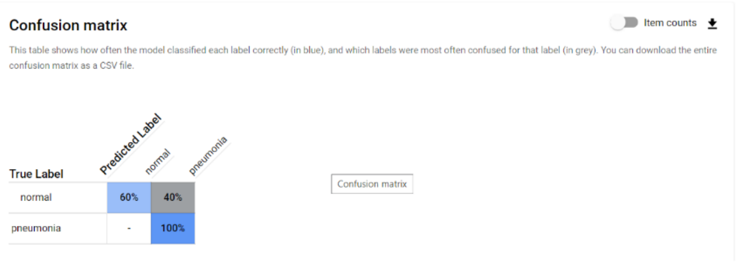
	(approaches 100%). ***Recall*** = TP/(TP+FN) = 0/(0+10) = 0/10 = 0 Consequently, for the normal class, we can say that precision tends to increase (to 100%), and recall tends to decrease (to 0%) for higher threshold values. ***Pneumonia class*** Like normal class precision tends to increase (to 100%), and recall tends to decrease (to 0%) for higher threshold values.

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>There were 400 images in total (100 normal & 300 pneumonia). From this dataset, 320 images (80 normal & 240 pneumonia) were used for training, 40 images were used for testing (10 normal & 30 pneumonia) and 40 images were used for validation (10 normal & 30 pneumonia) as shown in the image below:


You have enough images to start training
Unlabelled images aren't used. Your data set will be automatically split into Train, Validation and Test sets.
Ideally, each label should have at least 10 images. Fewer images often results in inaccurate precision and recall. You must also have at least 8, 1, 1 images each assigned to your Train, Validation and Test sets.
<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>
<tr>
<td><b>Confusion Matrix</b>
How has the confusion matrix been affected by the unbalanced data? Include a screenshot of the new confusion matrix.</td>
<td>With unbalanced data that have a majority of pneumonia images, prediction of normal images (minority class) has been affected negatively. As can be seen in the images below while 100% (10 out of 10) of predictions of normal images were correct for the balanced dataset, for the unbalanced dataset only 60% (6 out of 10) of normal images are correct. On the other hand, all the pneumonia images (30 in total, 100%) were predicted as pneumonia for the unbalanced dataset. Previously only 8 of 10 pneumonia images (80%) were predicted as pneumonia.

This is an expected result since the model tends to predict in favor of the majority class (pneumonia). If one class is much bigger in counts than the other class, the easiest way for the model is to predict every input as the majority class.</td>
</tr>
</table>



Confusion matrix

Item counts

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey). You can download the entire confusion matrix as a CSV file.

True Label \ Predicted Label	normal	pneumonia
normal	60%	40%
pneumonia	-	100%

Confusion matrix

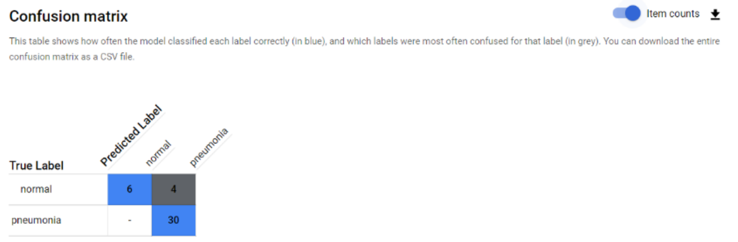
For ***normal*** class:

- TP = 6 (60%),
- FN = 4 (40%),
- FP = 0 (0%),
- TN = 30 (100%).

For ***pneumonia*** class:

- TP = 30 (100%),8
- FN = 0 (0%),2
- FP = 4 (40%),0
- TN = 6 (60%).10

We can conclude from the above results that for the ***normal*** class with unbalanced data TP and FP decrease and TN and FN increase. For the ***pneumonia*** class, on the other hand, TP and FP increase and TN and FN decrease.



Confusion matrix

Item counts

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey). You can download the entire confusion matrix as a CSV file.

True Label \ Predicted Label	normal	pneumonia
normal	6	4
pneumonia	-	30

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)?

Normal class

Precision = TP/(TP+FP) = 6/(6+0) = 6/6 = 1
Recall = TP/(TP+FN) = 6/(6+4) = 6/10 = 0,6

Pneumonia class

Precision = TP/(TP+FP) = 30/(30+4) = 30/34 ~= 0,88
Recall = TP/(TP+FN) = 30/(30+0) = 30/30 = 1

For the ***normal*** class *precision* increases from 0,83 to 1 and *recall* decreases from 1 to 0,6.

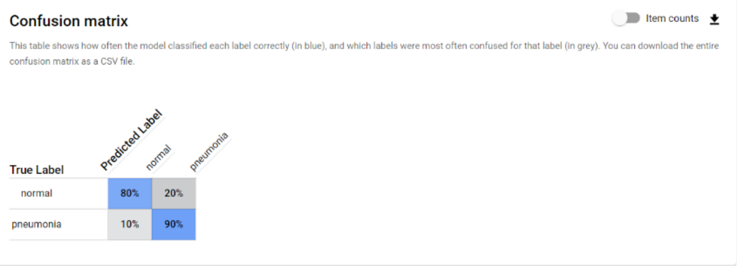
For the ***pneumonia*** class *precision* decreases from 1 to 0,88 and *recall* increases from 0,8 to 1.

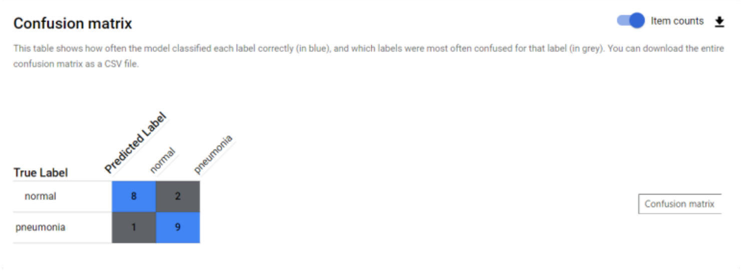
Unbalanced Classes
From what you have observed,

It is easily seen that a model trained with an unbalanced dataset favors the unbalanced class (Pneumonia for our case). In other

how do unbalanced classed affect a machine learning model?	words, the model tends to generalize the inputs to the majority class (Pneumonia). As a result, ***recall*** decreases for the minority class.

Binary Classifier with Dirty/Balanced Data

Confusion Matrix How has the confusion matrix been affected by the dirty data? Include a screenshot of the new confusion matrix.	As can be seen from the images below the confusion matrix is worse than the *clean/balanced* data case. This is completely expected. Because of the dirty data the training algorithm cannot learn the patterns in the data well enough.  **Confusion matrix** This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey). You can download the entire confusion matrix as a CSV file. True Label / Predicted Label: normal, pneumonia normal: 80%, 20% pneumonia: 10%, 90% for ***normal*** class: - TP = 8 (80%), - FN = 2 (20%), - FP = 1 (10%), - TN = 9 (90%). For ***pneumonia*** class: - TP = 9 (90%), - FN = 1 (10%), - FP = 2 (20%), - TN = 80 (80%). Although FP decreases from 2 to 1 and TN increases from 8 to 9, TP decreases from 10 to 8 and FN increases from zero to 2. The overall error (3 errors) is larger than the *clean/balanced* case (2 errors). In addition, there may be some dirty data in the test dataset which makes the model unreliable. This means that the real-life performance may be much worse than the evaluation.

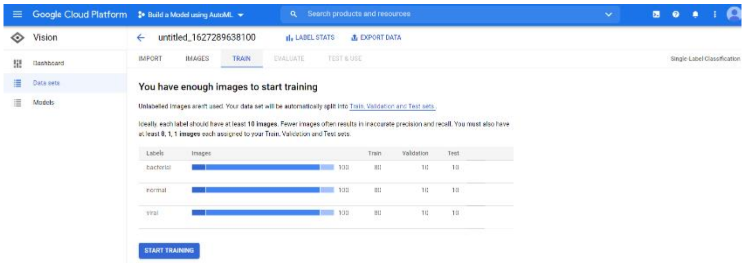
	 **Confusion matrix** Item counts This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey). You can download the entire confusion matrix as a CSV file. Predicted Label: normal, pneumonia True Label normal 8 2 pneumonia 1 9 Confusion matrix
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?	***Normal class*** ***Precision*** = TP/(TP+FP) = 8/(8+1) = 8/9 ~= 0,88 ***Recall*** = TP/(TP+FN) = 8/(8+2) = 8/10 = 0,8 ***Pneumonia class*** ***Precision*** = TP/(TP+FP) = 9/(9+2) = 9/11 ~= 0,81 ***Recall*** = TP/(TP+FN) = 9/(9+1) = 9/10 = 0,9 For the ***normal*** class *precision* increases from 0,83 to 0,88 and *recall* decreases from 1 to 0,8. For the ***pneumonia*** class *precision* decreases from 1 to 0,81 and *recall* increases from 0,8 to 0,9. For the ***normal*** class the *clean/unbalanced* classifier has the highest *precision* which is 1 (100%) and the *clean/balanced* classifier has the highest *recall* which is 1 (100%). For the ***pneumonia*** class the *clean/balanced* classifier has the highest *precision* which is 1 (100%) and the *clean/unbalanced* classifier has the highest *recall* which is 1 (100%).
Dirty Data From what you have observed, how does dirty data affect a machine learning model?	It is easily seen that the dirty data increases the errors for both classes. The sum of TP+TN has decreased, and the number or errors has increased with respect to *clean/balanced* case. Besides, If the data is dirty, we are not sure about the evaluation process itself. The results are unreliable because of the dirty data in the test and evaluation sets. The real-fife consequences may become much worse than the evaluation results.

3-Class Model

Confusion Matrix Summarize the 3-class confusion	There were 300 images in total (100 normal & 100 pneumonia). From this dataset, 240 images (80 normal, 80 viral pneumonia

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.

& 80 bacterial pneumonia) were used for training, 30 images were used for testing (10 normal, 10 viral pneumonia & 10 bacterial pneumonia) and 30 images were used for validation (10 normal, 10 viral pneumonia & 10 bacterial pneumonia) as shown in the image below:



You have enough images to start training

Unlabelled images aren't used. Your data set will be automatically split into Train, Validation and Test sets.

Ideally, each label should have at least **10 images**. Fewer images often results in inaccurate precision and recall. You must also have at least **8, 1, 1 images** each assigned to your Train, Validation and Test sets.

Labels	Images	Train	Validation	Test
bacterial	100	80	10	10
normal	100	80	10	10
viral	100	80	10	10

As can be seen from the images below, all the *normal* (10 in total) as well as *viral pneumonia* (10 in total) images were predicted as normal and *viral pneumonia* respectively. On the other hand, 4 of the *bacterial pneumonia* images (10 in total) were predicted as *viral pneumonia* and 6 of the *bacterial pneumonia* images were predicted as *bacterial pneumonia*.

From the images below we can see that:

For ***normal*** class:

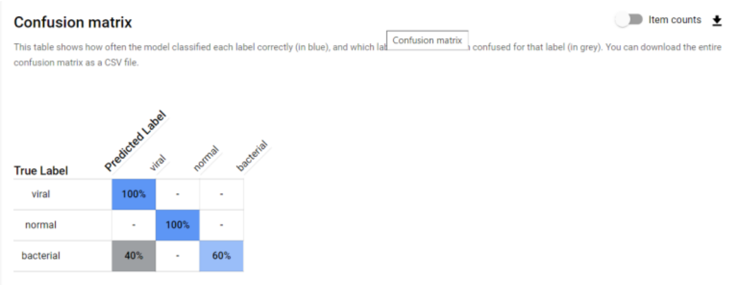
- TP = 10 (100%),
- FN = 0 (0%),
- FP = 0 (0%),

For ***viral pneumonia*** class:

- TP = 10 (100%),
- FN = 0 (0%),
- FP = 4 (40%),

For ***bacterial pneumonia*** class:

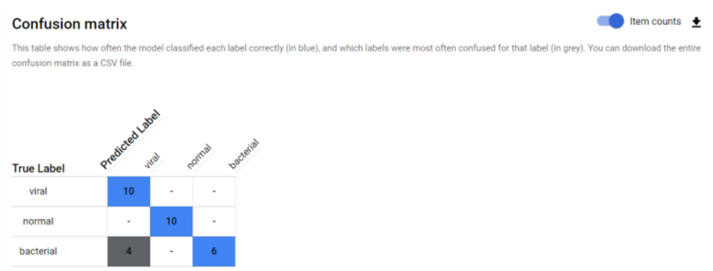
- TP = 6 (60%),
- FN = 4 (40%),
- FP = 0 (0%),



Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which lal Confusion matrix confused for that label (in grey). You can download the entire confusion matrix as a CSV file.

True Label	Predicted Label viral	normal	bacterial
viral	100%	-	-
normal	-	100%	-
bacterial	40%	-	60%



Confusion matrix

Item counts

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey). You can download the entire confusion matrix as a CSV file.

True Label \ Predicted Label	viral	normal	bacterial
viral	10	-	-
normal	-	10	-
bacterial	4	-	6

The model seems to confuse *bacterial pneumonia* with *viral pneumonia*. The *normal* class seems to be OK. We can add new data to remove the confusion. For the dataset to be balanced we need to add equal data for every class. I have added 20 images for each class and checked the results. Since the amount of data for each class increased 20%, the number of test images increased 20% also and become 12 images for each data.

As can be seen from the images below:

For ***normal*** class:

- TP = 12 (100%),
- FN = 0 (0%),
- FP = 0 (0%),

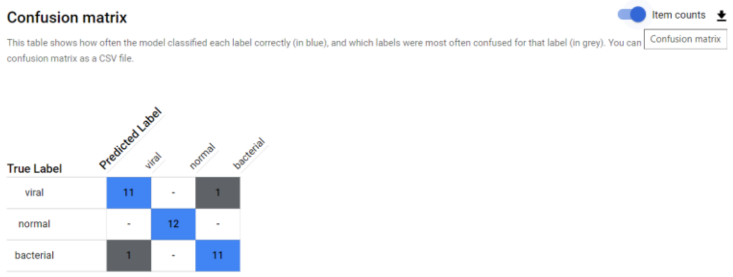
For ***viral pneumonia*** class:

- TP = 11 (92%),
- FN = 1 (8%),
- FP = 1 (8%),

For ***bacterial pneumonia*** class:

- TP = 11 (92%),
- FN = 1 (8%),
- FP = 1 (8%),

The calculations above shows that we have a better model with the bigger dataset. We can also see this by calculating *precision*, *recall* and *F1 score* below.

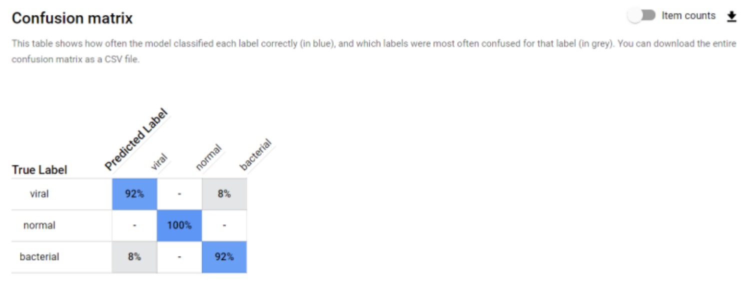


Confusion matrix

Item counts

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey). You can Confusion matrix confusion matrix as a CSV file.

True Label \ Predicted Label	viral	normal	bacterial
viral	11	-	1
normal	-	12	-
bacterial	1	-	11



Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in grey). You can download the entire confusion matrix as a CSV file.

True Label \ Predicted Label	viral	normal	bacterial
viral	92%	-	8%
normal	-	100%	-
bacterial	8%	-	92%

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)?

AutoML with 300 images:

Normal class

Precision = TP/(TP+FP) = 10/(10+0) = 10/10 = 1
Recall = TP/(TP+FN) = 10/(10+0) = 10/10 = 1

Viral Pneumonia class

Precision = TP/(TP+FP) = 10/(10+4) = 10/14 ~= 0,71
Recall = TP/(TP+FN) = 10/(10+0) = 10/10 = 1

Bacterial Pneumonia class

Precision = TP/(TP+FP) = 6/(6+0) = 6/6 = 1
Recall = TP/(TP+FN) = 6/(6+4) = 6/10 = 0,6

AutoML with 360 images:

Normal class

Precision = TP/(TP+FP) = 12/(12+0) = 12/12 = 1
Recall = TP/(TP+FN) = 12/(12+0) = 12/12 = 1

Viral Pneumonia class

Precision = TP/(TP+FP) = 11/(11+1) = 11/12 ~= 0,92
Recall = TP/(TP+FN) = 11/(11+1) = 11/12 ~= 0,92

Bacterial Pneumonia class

Precision = TP/(TP+FP) = 11/(11+1) = 11/12 ~= 0,92
Recall = TP/(TP+FN) = 11/(11+1) = 11/12 ~= 0,92

F1 Score
What is this model's F1 score?

F1 score = $2 * (P_m*R_m)/(P_m+R_m)$

AutoML with 300 images:

Normal class

F1 = 2 * (1*1) / (1+1) = 2 * (1/2) = 1

	Viral Pneumonia class ***F1*** = 2 * (0,71*1) / (0,71+1) = 2 * (0,71/1,71) ~= 0,83 ***Bacterial Pneumonia class*** ***F1*** = 2 * (1*0,6) / (1+0,6) = 2 * (0,6/1,6) = 0,75 ***AutoML with 360 images:*** ***Normal class*** ***F1*** = 2 * (1*1) / (1+1) = 2 * (1/2) = 1 ***Viral Pneumonia class*** ***F1*** = 2 * (0,92*0,92) / (0,92+0,92) = 2 * (0,8464/1,84) ~= 0,92 ***Bacterial Pneumonia class*** ***F1*** = 2 * (0,92*0,92) / (0,92+0,92) = 2 * (0,8464/1,84) ~= 0,92