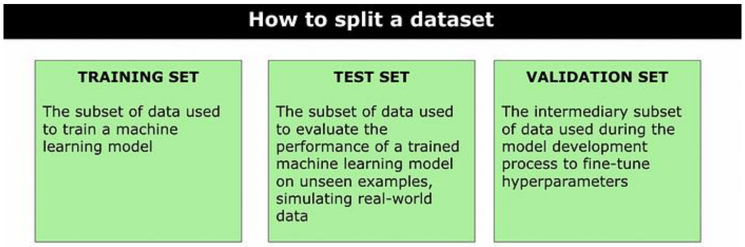
AutoML Modeling Report

<Ugur Uresin, December 16, 2023>

Binary Classifier with Clean/Balanced Data

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

*



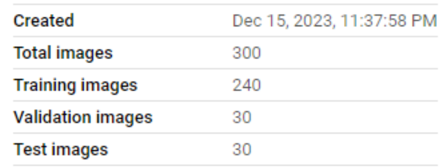
How to split a dataset

TRAINING SET
The subset of data used to train a machine learning model

TEST SET
The subset of data used to evaluate the performance of a trained machine learning model on unseen examples, simulating real-world data

VALIDATION SET
The intermediary subset of data used during the model development process to fine-tune hyperparameters

300 random images in total were selected from the Kaggle's dataset (150 normal & 150 pneumonia).
From this dataset, 120 images (60 normal & 60 pneumonia) were used for training, 30 images were used for testing (15 normal & 15 pneumonia) and 30 images were used for validation (15 normal & 15 pneumonia) as shown in the image below:

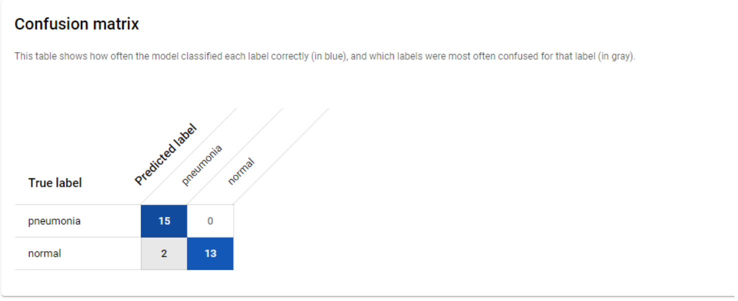


Created	Dec 15, 2023, 11:37:58 PM
Total images	300
Training images	240
Validation images	30
Test images	30

* Reference:
https://medium.com/syntaxerrorpub/understanding-the-difference-between-training-test-and-validation-sets-in-machine-learning-c59feec6483b

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?

The confusion matrix in the image is a tool used in machine learning to evaluate the performance of classification models. It shows the number of correct and incorrect predictions made by the model, compared to the actual labels.



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 gray).

True label	Predicted label pneumonia	normal
pneumonia	15	0
normal	2	13

Here's what each cell represents:

True Positive (TP): The model correctly predicted the positive class (in this case, "pneumonia"). Here, the TP for pneumonia is 15.

True Negative (TN): The model correctly predicted the negative class (in this case, "normal"). The TN for normal is 13.

False Positive (FP): The model incorrectly predicted the positive class. Here, there are 0 instances where the model predicted pneumonia when it was actually normal.

False Negative (FN): The model incorrectly predicted the negative class. Here, there are 2 instances where the model predicted normal when it was actually pneumonia.

Given these values, we can calculate the following:
True Positive Rate (TPR) for the "pneumonia" class, also known as sensitivity or recall, is TP / (TP + FN).

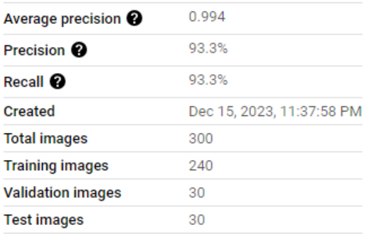
So it would be 15 / (15 + 2) = 15 / 17.

False Positive Rate (FPR) for the "normal" class is FP / (FP + TN), which would be 0 / (0 + 13) = 0,
since there are no false positives for the normal class.

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 the accuracy of positive predictions. It is defined as TP / (TP + FP), the proportion of positive identifications that were actually correct.

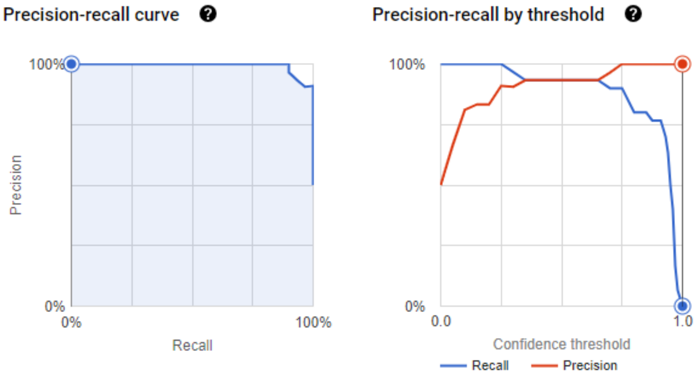
Recall measures the ability of a model to find all the relevant cases within a dataset. It is defined as TP / (TP + FN), the proportion of actual positives that were correctly identified.



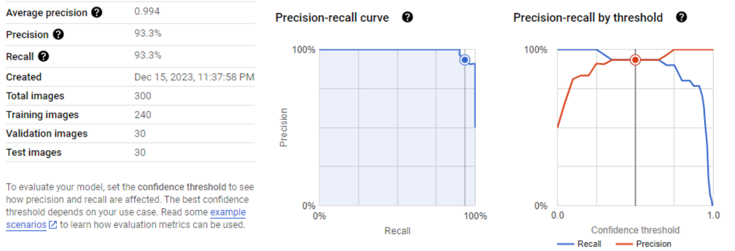
Average precision ?	0.994
Precision ?	93.3%
Recall ?	93.3%
Created	Dec 15, 2023, 11:37:58 PM
Total images	300
Training images	240
Validation images	30
Test images	30

For this model, the precision and recall at the score threshold of 0.5 for the "pneumonia" class are:

Precision: It cannot be calculated directly from the confusion matrix because FP for the "normal" class is zero, and we do not have the number of false positives for the "pneumonia" class. If we assume there are no false positives for "pneumonia" as well, the precision would be 100%



Recall: As given below, the recall for "pneumonia" is 15 / 17 (93.3%)

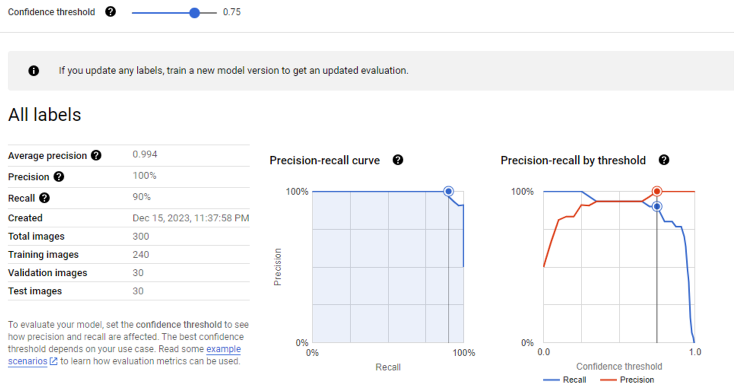


Average precision ❓	0.994
Precision ❓	93.3%
Recall ❓	93.3%
Created	Dec 15, 2023, 11:37:58 PM
Total images	300
Training images	240
Validation images	30
Test images	30

To evaluate your model, set the **confidence threshold** to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some example scenarios to learn how evaluation metrics can be used.

Score Threshold

When you increase the threshold what happens to precision? What happens to recall? Why?



Confidence threshold ❓ 0.75

ℹ If you update any labels, train a new model version to get an updated evaluation.

All labels

Average precision ❓	0.994
Precision ❓	100%
Recall ❓	90%
Created	Dec 15, 2023, 11:37:58 PM
Total images	300
Training images	240
Validation images	30
Test images	30

To evaluate your model, set the **confidence threshold** to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some example scenarios to learn how evaluation metrics can be used.

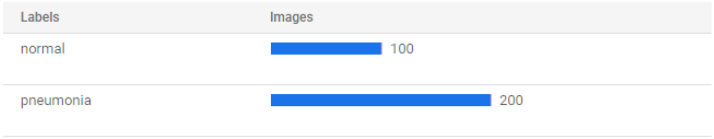
When the threshold is set to 0.75, the precision of the model is at 100%.
The recall at this threshold is lower than the precision, though the exact value isn't visible in the graph. It appears to be less than 100% but higher than the recall at a threshold of 1.0.

	The precision and recall are measures of a model's accuracy and are affected differently by the confidence threshold: Precision is the ratio of true positives to the total number of positive predictions made by the model. A higher confidence threshold generally leads to a higher precision because the model is more conservative in making positive predictions; it only makes a prediction when it is very sure. This means it makes fewer positive predictions, but those predictions are more likely to be correct. Recall is the ratio of true positives to the actual number of positive instances in the dataset. As the confidence threshold increases, the model requires higher confidence to make a positive prediction, which means it may miss some actual positives, leading to a lower recall. It's a measure of how many of the actual positives the model is able to capture. In summary, as you increase the confidence threshold: Precision may remain high or even increase because the model makes fewer false positive predictions. Recall tends to decrease because the model may miss more true positive cases, as it is more stringent in its prediction criteria. This is a trade-off often encountered in classification tasks: increasing precision typically comes at the cost of decreasing recall, and vice versa. The ideal balance depends on the specific application and the cost associated with false positives versus false negatives.

Binary Classifier with Clean/Unbalanced Data

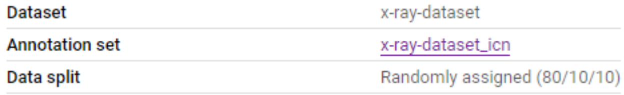
Train/Test Split
How much data was used for training? How much data was used for testing?

300 images were used. 100 for normal, 200 for pneumonia as follows:



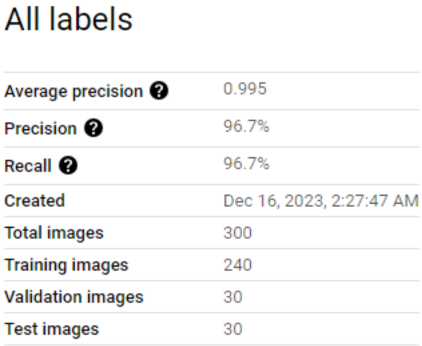
Labels	Images
normal	100
pneumonia	200

The split ratio is 80/10/10 for training/validation/test as follows:



Dataset	x-ray-dataset
Annotation set	x-ray-dataset_icn
Data split	Randomly assigned (80/10/10)

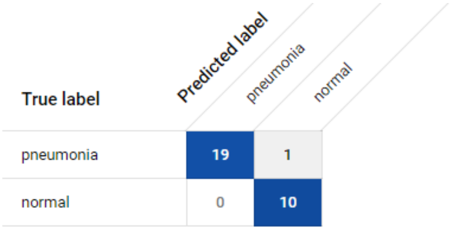
240 images for training, 30 images for validation, 30 images for testing



All labels

Average precision ?	0.995
Precision ?	96.7%
Recall ?	96.7%
Created	Dec 16, 2023, 2:27:47 AM
Total images	300
Training images	240
Validation images	30
Test images	30

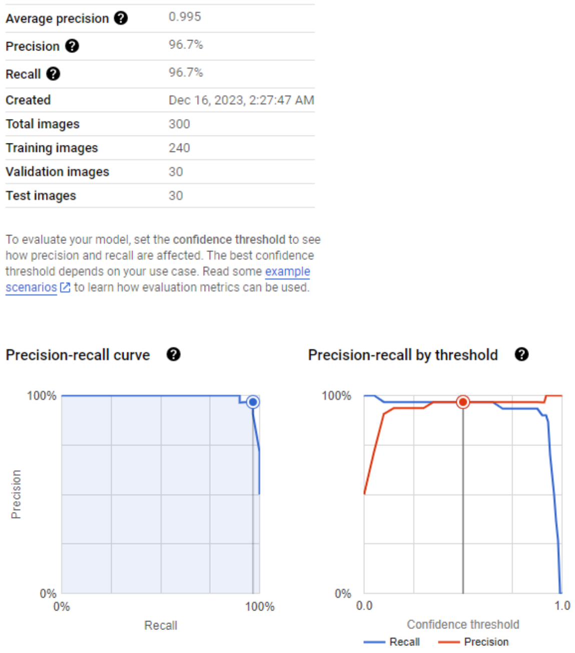
Confusion Matrix
How has the confusion matrix been affected by the unbalanced data? Include a screenshot of the new confusion matrix.



True label	**Predicted label** pneumonia	normal
pneumonia	19	1
normal	0	10

The imbalance can lead to a bias in the model where it may become better at identifying "pneumonia" simply because it has more examples of that class to learn from.

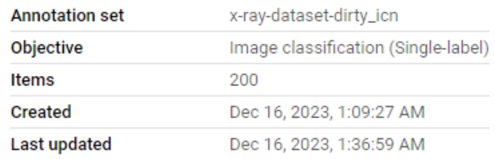
From the confusion matrix:
The model predicted "pneumonia" correctly 19 times and incorrectly only once when the true label was "normal".
It predicted "normal" correctly 10 times, and there are no instances where "normal" was predicted when the true label was "pneumonia".
Given the unbalanced dataset, the model might be overfitting to the "pneumonia" class, which could explain the high performance in predicting "pneumonia" correctly. The lack of false positives for "normal" is a good sign, indicating that when the model predicts "normal", it is highly likely to be correct.

<table>
<tr>
<td><b>Precision and Recall</b>
How have the model’s precision and recall been affected by the unbalanced data (report the values for a score threshold of 0.5)?</td>
<td>

<table>
<tr><td>Average precision ❓</td><td>0.995</td></tr>
<tr><td>Precision ❓</td><td>96.7%</td></tr>
<tr><td>Recall ❓</td><td>96.7%</td></tr>
<tr><td>Created</td><td>Dec 16, 2023, 2:27:47 AM</td></tr>
<tr><td>Total images</td><td>300</td></tr>
<tr><td>Training images</td><td>240</td></tr>
<tr><td>Validation images</td><td>30</td></tr>
<tr><td>Test images</td><td>30</td></tr>
</table>
To evaluate your model, set the confidence threshold to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some example scenarios to learn how evaluation metrics can be used.

The precision and recall of the model are both 96.7%. The average precision across all thresholds is 0.995.
</td>
</tr>
<tr>
<td><b>Unbalanced Classes</b>
From what you have observed, how do unbalanced classed affect a machine learning model?</td>
<td>These high values for precision and recall suggest that the model is performing well on the test set. However, considering that the data is unbalanced, with twice as many "pneumonia" images as "normal", this could potentially lead to a bias in the model.
Moreover, if the model is overly confident in predicting the majority class, it could ignore the minority class, which would lead to a high number of false negatives for the minority class, thus affecting the recall.</td>
</tr>
</table>

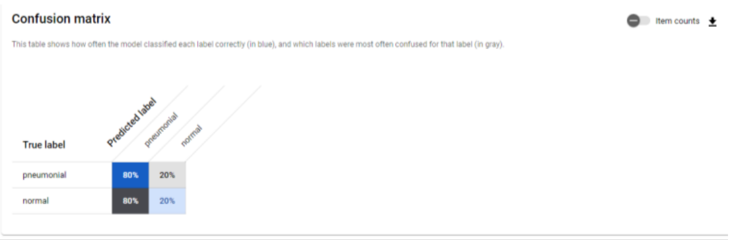
Binary Classifier with Dirty/Balanced Data

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



Annotation set	x-ray-dataset-dirty_icn
Objective	Image classification (Single-label)
Items	200
Created	Dec 16, 2023, 1:09:27 AM
Last updated	Dec 16, 2023, 1:36:59 AM

A dirty dataset was created by intentionally mislabeling 30% of the images for both labels (normal, pneumonia). A total of 200 images were used.

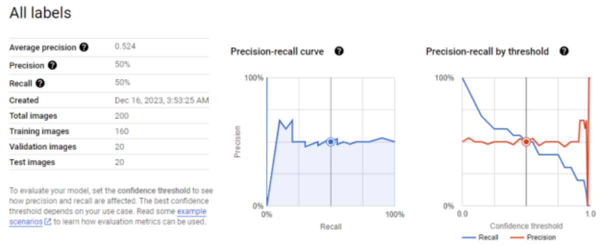


Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray).

True label	Predicted label: pneumonial	normal
pneumonial	80%	20%
normal	80%	20%

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?



All labels

Average precision	0.524
Precision	50%
Recall	50%
Created	Dec 16, 2023, 3:53:25 AM
Total images	200
Training images	160
Validation images	20
Test images	20

To evaluate your model, set the **confidence threshold** to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some example scenarios to learn how evaluation metrics can be used.

Both precision and recall are at 50%.
The average precision across all thresholds is 0.524.

For a score threshold of 0.5, if we assume the values shown are at this default threshold, the precision and recall are equal at 50%.

This typically indicates that the model is performing no better than random guessing for this threshold, which is a common occurrence when dealing with dirty data.

As for which binary classifier has the highest precision and recall:

The confusion matrix shows that both classes ("pneumonia" and "normal") have the same precision and recall.

Hence, both classifiers have equal precision and recall at 80% for correctly identifying true labels and 20% for mislabeling the other class.

However, if we consider the precision and recall at the threshold of 0.5 (assuming the provided values are for this threshold), both precision and recall are at 50%.

Dirty Data
From what you have observed, how does dirty data affect a machine learning model?

Dirty data typically refers to data that is incorrect, misleading, or poorly formatted. This can include mislabeled examples, as in your case, where 30% of the images for both classes were intentionally mislabeled. Dirty data can affect machine learning models in several ways:

Decreased Accuracy:
Mislabeled data can lead to a model learning incorrect patterns, decreasing its overall accuracy.

Overfitting:
The model may overfit to noise and errors in the training data, leading to poor generalization to new, unseen data.

Skewed Metrics:
Performance metrics can become skewed, as the model may appear to perform well on the dirty data but fail on clean data.

Increased Uncertainty:
Dirty data increases the uncertainty of predictions and may require additional post-processing or manual review to ensure the quality of the model's outputs.

Misguided Feature Importance:
The model may assign importance to features that are related to the noise in the dirty data, rather than to the true underlying patterns that are predictive of the correct outcomes.

An Experience from an Industrial AI Use Case
I have had an experience with dirty data where we used incorrect labels while training the model for an error mode that was unlikely to occur.

This particular error mode did not occur in the business unit for a long time, leading us to believe that our model was performing well, as we did not observe any data drift.

However, when this low-probability but high-impact error mode began to occur in the business unit, we realized that the model failed to catch it for a considerable period. An error analysis revealed that the initial data was not correctly labeled for this specific error mode.

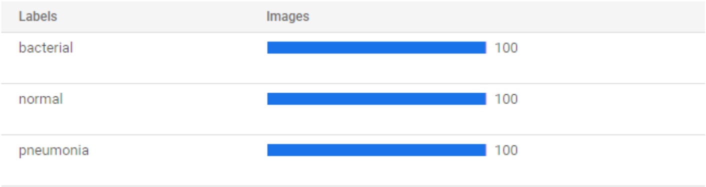
Since it was an industrial artificial intelligence application, it resulted in a loss of time for the business unit.

Therefore, it's very important to align with the business unit from the beginning to ensure that the labels are accurate.

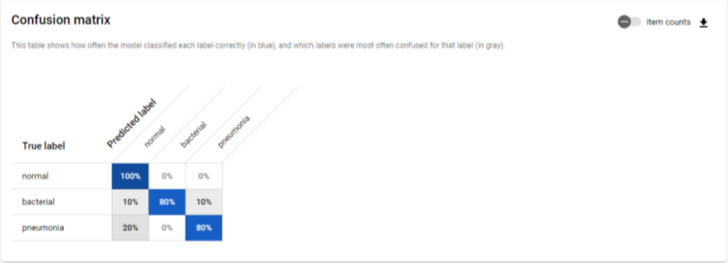
3-Class Model

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.

300 images were used and there are equally distributed 3 labels as follows:



Labels	Images
bacterial	100
normal	100
pneumonia	100



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 gray).

True label	Predicted label normal	bacterial	pneumonia
normal	100%	0%	0%
bacterial	10%	80%	10%
pneumonia	20%	0%	80%

Normal: The model performs perfectly on this class, with 100% accuracy. No normal cases are confused with either bacterial or pneumonia.

Bacterial: The model correctly identifies 80% of bacterial cases, but confuses 10% of bacterial cases as normal and another 10% as pneumonia.

Pneumonia: The model correctly identifies 80% of pneumonia cases, but confuses 20% of pneumonia cases as normal.

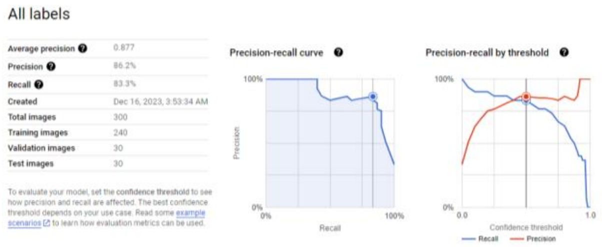
Potential Remedies for Model Confusion:

Data Augmentation: You could increase the number of training examples for bacterial and pneumonia classes, especially focusing on cases that are typical for each class.

Feature Engineering: Identify and include more discriminative features that can help distinguish between bacterial and pneumonia cases.

Error Analysis: Perform an in-depth analysis of the misclassified cases to understand why the model is confusing them, and then take corrective measures based on the findings.
Model Complexity: Increase the complexity of the model if it is too simple and unable to capture the nuances between the classes, or consider using a different model architecture.

Class Weighting: Adjust the weights of classes during training to penalize the model more for confusing bacterial and pneumonia cases.

<table>
<tr>
<td><b>Precision and Recall</b>
What are the model’s precision and recall? How are these values calculated (report the values for a score threshold of 0.5)?</td>
<td>


All labels

Average precision ❓	0.877
Precision ❓	86.2%
Recall ❓	83.3%
Created	Dec 16, 2023, 3:53:34 AM
Total images	300
Training images	240
Validation images	30
Test images	30

To evaluate your model, set the **confidence threshold** to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some example scenarios to learn how evaluation metrics can be used.

For Normal:
True Positives (TP_normal): 100%
False Positives (FP_normal): 0% for bacterial + 0% for pneumonia
False Negatives (FN_normal): 0%

For Bacterial:
True Positives (TP_bacterial): 80%
False Positives (FP_bacterial): 10% for normal + 0% for pneumonia
False Negatives (FN_bacterial): 10% for normal + 10% for pneumonia

For Pneumonia:
True Positives (TP_pneumonia): 80%
False Positives (FP_pneumonia): 0% for normal + 20% for bacterial
False Negatives (FN_pneumonia): 20% for normal + 0% for bacterial
</td>
</tr>
<tr>
<td><b>F1 Score</b>
What is this model’s F1 score?</td>
<td>F1=2×(PrecisionxRecall) / (Precision+Recall)
The F1 score for the model, given the precision of 86.2% and recall of 83.3%, is approximately 0.847 or 84.73%</td>
</tr>
</table>