AutoML Modeling Report

Mariyyah Essam Samarin

Binary Classifier with Clean/Balanced Data

Train/Test Split How much data was used for training? How much data was used for testing?	Training data used to build model, the model will learn how to distinguish between models based on these data, while the testing data used to evaluate model performance. In addition, if we want to avoid model peeking during hyperparameter tuning we need to use validation set. In our case we have 198 images and we divide it into 158 JPEG X-Ray images for training data (0.80) and 40 images for testing data (0.20), both sets have balanced amount of (Pneumonia/Normal) samples.

Confusion Matrix
What do each of the sections 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?
	{
	"AggregatedEvaluationResults": {
	"ConfusionMatrix": [
	{
	"GroundTruthLabel": "normal",
	"PredictedLabel": "normal",
	"Value": 0.95
	},
	{
	"GroundTruthLabel": "normal",
	"PredictedLabel": "pneumonia",
	"Value": 0.05
	},
	{
	"GroundTruthLabel": "pneumonia",
	"PredictedLabel": "normal",
	"Value": 0.0
	},
	{
	"GroundTruthLabel": "pneumonia",
	"PredictedLabel": "pneumonia",
	"Value": 1.0
	}
	],
	"F1Score": 1.0,
	"Precision": 1.0,
	"Recall": 1.0
	},
	First, we need to define null hypothesis which is the patient have pneumonia, now it is easy to define sections of confusion matrix:
	1-True Positive (0.95): reject the null hypothesis, when the fact is true.

	2-True Negative (0): failing to reject the null hypothesis when the fact is false. 3-False positive or type 1 error (0.05): reject the null hypothesis, when the fact is false. 4- False Negative or type 2 error (0): failing to reject the null hypothesis when the fact is true. We can notice the overfitting of the model because is learn the noise -TP rate = Recall = 1 -FP rate = 0.05 / (0.05 + 0) = 1
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 measure the ratio of retrieved samples that are relevant, meanwhile recall measure the relevant samples that are retrieved among all dataset. Both of recall and precision are equal to 1 which indicates the model is accurate.

Binary Classifier with Clean/Unbalanced Data

Train/Test Split How much data was used for training? How much data was used for testing?	Dataset:400 images Pneumonia class:300 images Normal class:100 images Hold-Out strategy: 80% for training and 20% for testing Training data: 320 images Testing data: 80 images

Confusion Matrix
How has the confusion matrix been affected by the unbalanced data? Include a screenshot of the new confusion matrix summary
	{
	"AggregatedEvaluationResults": {
	"ConfusionMatrix": [
	{
	"GroundTruthLabel": "normal",
	"PredictedLabel": "normal",
	"Value": 0.55
	},
	{
	"GroundTruthLabel": "normal",
	"PredictedLabel": "pneumonia",
	"Value": 0.45
	},
	{
	"GroundTruthLabel": "pneumonia",
	"PredictedLabel": "normal",
	"Value": 0.0
	},
	{
	"GroundTruthLabel": "pneumonia",
	"PredictedLabel": "pneumonia",
	"Value": 1.0
	}
	],
	"F1Score": 1.0,
	"Precision": 1.0,
	"Recall": 1.0
	},

	Compare to clean and balanced dataset, the TP decrease (from 0.95 to 0.55) while FP increase (from 0.05 to 0.45), that mean when we add more examples of pneumonia class , so we will have more pneumonia examples in training and testing sets .
Precision and Recall How have the model's precision and recall been affected by the unbalanced data?	Model's precision and recall does not change according to the confusion matrix, but in general it may be affected. However, if we have unbalance dataset, we need to use other evaluation method such as ROC or cross validation.
Unbalanced Classes From what you have observed, how do unbalanced classed affect a machine learning model?	The model will bias toward majority class, it will always classify the sample as the class with most examples which mean the model cannot generalize on unseen data.

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 information.
	{
	"AggregatedEvaluationResults": {
	"ConfusionMatrix": [
	{
	"GroundTruthLabel": "normal",
	"PredictedLabel": "normal",
	"Value": 0.75
	},
	{
	"GroundTruthLabel": "normal",
	"PredictedLabel": "pneumonia",
	"Value": 0.25
	},
	{
	"GroundTruthLabel": "pneumonia",
	"PredictedLabel": "normal",
	"Value": 0.2
	},
	{
	"GroundTruthLabel": "pneumonia",
	"PredictedLabel": "pneumonia",
	"Value": 0.8
	}
	],
	"F1Score": 0.808392315470172,
	"Precision": 0.7349498327759196,
	"Recall": 0.8999999999999999
	},

	Type 1 errors or false alarm increase, but we are more curious about type 2 errors when diagnosing pneumonia because of the risk of classifying a patient as healthy while he/she is sick. Type 2 error relates to false negative which is 0.3 here.
Precision and Recall How have the model's precision and recall been affected by the dirty data. Of the binary classifiers, which has the highest precision? Which has the highest recall?	Precision and recall relate to false positive and false negative respectively. So, we can see the changes in recall and precision because changing in FP and FN amount. highest precision=highest recall=1 when we have clean data.
Dirty Data From what you have observed, how does dirty data affect a machine learning model?	If the model learns from dirty data, leads to the wrong classification which reduces the performance and generalization 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 information.
	{
	"AggregatedEvaluationResults": {
	"ConfusionMatrix": [
	{
	"GroundTruthLabel": "bacterial_pneumonia",
	"PredictedLabel": "bacterial_pneumonia",
	"Value": 0.9
	},
	{
	"GroundTruthLabel": "bacterial_pneumonia",
	"PredictedLabel": "normal",
	"Value": 0.0
	},
	{
	"GroundTruthLabel": "bacterial_pneumonia",
	"PredictedLabel": "viral_pneumonia",
	"Value": 0.1
	},
	{
	"GroundTruthLabel": "normal",
	"PredictedLabel": "bacterial_pneumonia",
	"Value": 0.05
	},
	{
	"GroundTruthLabel": "normal",
	"PredictedLabel": "normal",
	"Value": 0.75
	},
	{
	"GroundTruthLabel": "normal",
	"PredictedLabel": "viral_pneumonia",
	"Value": 0.2
	},

<table>
<tr>
<td></td>
<td>
<pre>
  {
    "GroundTruthLabel": "viral_pneumonia",
    "PredictedLabel": "bacterial_pneumonia",
    "Value": 0.1
  },
  {
    "GroundTruthLabel": "viral_pneumonia",
    "PredictedLabel": "normal",
    "Value": 0.0
  },
  {
    "GroundTruthLabel": "viral_pneumonia",
    "PredictedLabel": "viral_pneumonia",
    "Value": 0.9
  }
],
"F1Score": 0.9258669930640555,
"Precision": 0.9231884057971014,
"Recall": 0.9333333333333332
</pre>

Which classes is the model most likely to confuse?
It seems the model confused between bacterial and viral pneumonia, about 10 percent of viral pneumonia is classified as bacterial pneumonia. In addition, there is about 20% of normal images classified as viral pneumonia which indicates that the model cannot distinguish well between these two models.

Which class(es) is the model most likely to get right?
The class normal has most TP rate.

Why might you do to try to remedy the model's "confusion"?
Add more images of both classes so the model can learn more about data.
</td>
</tr>
<tr>
<td>Precision and Recall
What are the model's precision and recall? How are these values calculated?</td>
<td>Once have a 3-class classifier we will calculate Precision and Recall for each class and then take the average to be an overall performance measure.
These are the formulas that should be applied for each class:
Precision =TP / (TP + FP)
Recall = TP / (TP + FN)
Then we generalize the measurement over the whole model:
Precision = Sum of precision for each class / No. of class = 0.9231884 = 92.3%
Recall = Sum of recall for each class / No. of class = 0.9333…= 93.3%</td>
</tr>
<tr>
<td>F1 Score
What is this model's F1 score?</td>
<td>F1 score is used with the unbalanced dataset to evaluate model performance and take precision and recall into</td>
</tr>
</table>

	account, an F1 score that is closer to one has better performance. F1 score= 0.92586, as we have 3-class the F1 score is the average of F1 score among all classes. F1 score = Sum of F1 score for each class / No. of class