Evaluation

Introduction

Till now we have learnt about the 4 stages of AI project cycle, viz. Problem scoping, Data acquisition, Data exploration and modelling. While in modelling we can make different types of models, how do we check if one's better than the other? That's where Evaluation comes into play. In the Evaluation stage, we will explore different methods of evaluating an AI model. Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future

What is evaluation?

Evaluation is the process of understanding the reliability of any AI model, based on outputs by feeding test dataset into the model and comparing with actual answers. There can be different Evaluation techniques, depending of the type and purpose of the model. Remember that It's not recommended to use the data we used to build the model to evaluate it. This is because our model will simply remember the whole training set, and will therefore always predict the correct label for any point in the training set. This is known as overfitting.

Firstly, let us go through various terms which are very important to the evaluation process.

Model Evaluation Terminologies

There are various new terminologies which come into the picture when we work on evaluating our model. Let's explore them with an example of the Forest fire scenario.

The Scenario

Imagine that you have come up with an AI based prediction model which has been deployed in a forest which is prone to forest fires. Now, the objective of the model is to predict whether a forest fire has broken out in the forest or not. Now, to understand the efficiency of this model, we need to check if the predictions which it makes are correct or not. Thus, there exist two conditions which we need to ponder upon: Prediction and Reality. The prediction is the output which is given by the machine and the reality is the real scenario in the forest when the prediction has been made. Now let us look at various combinations that we can have with these two conditions.

Case 1: Is there a forest fire?

Prediction: Yes

Reality: Yes

True Positive

Here, we can see in the picture that a forest fire has broken out in the forest. The model predicts a Yes which means there is a forest fire. The Prediction matches with the Reality. Hence, this condition is termed as **True Positive**.

Case 2: Is there a forest fire?

Prediction: No

Reality: No

True Negative

Here there is no fire in the forest hence the reality is No. In this case, the machine too has predicted it correctly as a No. Therefore, this condition is termed as **True Negative**.

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Case 3: Is there a forest fire?

Prediction: Yes

Reality: No

False Positive

Here the reality is that there is no forest fire. But the machine has incorrectly predicted that there is a forest fire. This case is termed as **False Positive**.

Case 4: Is there a forest fire?

Prediction: No

Reality: Yes

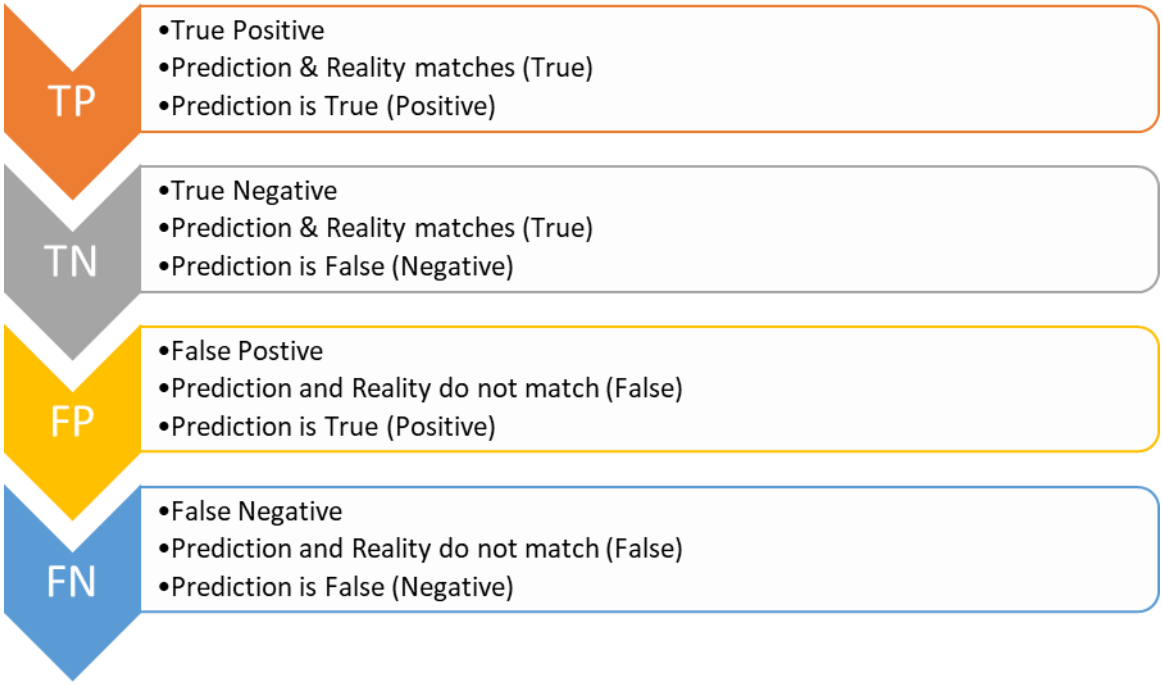
False Negative

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Here, a forest fire has broken out in the forest because of which the Reality is Yes but the machine has incorrectly predicted it as a No which means the machine predicts that there is no Forest Fire. Therefore, this case becomes **False Negative**.

Confusion matrix

The result of comparison between the prediction and reality can be recorded in what we call the confusion matrix. The confusion matrix allows us to understand the prediction results. Note that it is not an evaluation metric but a record which can help in evaluation. Let us once again take a look at the four conditions that we went through in the Forest Fire example:



Let us now take a look at the confusion matrix:

The Confusion Matrix		Reality	
		Yes	**No**
Prediction	**Yes**	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

Prediction and Reality can be easily mapped together with the help of this confusion matrix.

Evaluation Methods

Now as we have gone through all the possible combinations of Prediction and Reality, let us see how we can use these conditions to evaluate the model.

Accuracy

Accuracy is defined as the percentage of correct predictions out of all the observations. A prediction can be said to be correct if it matches the reality. Here, we have two conditions in which the Prediction matches with the Reality: True Positive and True Negative. Hence, the formula for Accuracy becomes:

$$Accuracy = \frac{Correct\ prediction}{Total\ cases} * 100\%$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} * 100\%$$

Here, total observations cover all the possible cases of prediction that can be True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

As we can see, Accuracy talks about how true the predictions are by any model. Let us ponder:

Is high accuracy equivalent to good performance?

_____
_____

How much percentage of accuracy is reasonable to show good performance?

_____
_____

Let us go back to the Forest Fire example. Assume that the model always predicts that there is no fire. But in reality, there is a 2% chance of forest fire breaking out. In this case, for 98 cases, the model will be right but for those 2 cases in which there was a forest fire, then too the model predicted no fire.

Here,

True Positives = 0

True Negatives = 98

Total cases = 100

Therefore, accuracy becomes: (98 + 0) / 100 = 98%

This is a fairly high accuracy for an AI model. But this parameter is useless for us as the actual cases where the fire broke out are not taken into account. Hence, there is a need to look at another parameter which takes account of such cases as well.

Precision

Precision is defined as the percentage of true positive cases versus all the cases where the prediction is true. That is, it takes into account the True Positives and False Positives.

$$Precision = \frac{True\ Positive}{All\ Predicted\ Positives} * 100\%$$

$$Precision = \frac{TP}{TP + FP} * 100\%$$

Going back to the Forest Fire example, in this case, assume that the model always predicts that there is a forest fire irrespective of the reality. In this case, all the Positive conditions would be taken into account that is, True Positive (Prediction = Yes and Reality = Yes) and False Positive (Prediction = Yes and Reality = No). In this case, the firefighters will check for the fire all the time to see if the alarm was True or False.

You might recall the story of the boy who falsely cries out that there are wolves every time and so when they actually arrive, no one comes to his rescue. Similarly, here if the Precision is low (which means there are more False alarms than the actual ones) then the firefighters would get complacent and might not go and check every time considering it could be a false alarm.

This makes Precision an important evaluation criteria. If Precision is high, this means the True Positive cases are more, giving lesser False alarms.

But again, is good Precision equivalent to a good model performance? Why?

_____
_____
_____
_____
_____
_____

Prediction: 10 cases of TP

Reality: 20 cases of yes

100% precise
But is it usable?

Let us consider that a model has 100% precision. Which means that whenever the machine says there's a fire, there is actually a fire (True Positive). In the same model, there can be a rare exceptional case where there was actual fire but the system could not detect it. This is the case of a False Negative condition. But the precision value would not be affected by it because it does not take FN into account. Is precision then a good parameter for model performance?

Recall

Another parameter for evaluating the model's performance is Recall. It can be defined as the fraction of positive cases that are correctly identified. It majorly takes into account the true reality cases where in Reality there was a fire but the machine either detected it correctly or it didn't. That is, it considers True Positives (There was a forest fire in reality and the model predicted a forest fire) and False Negatives (There was a forest fire and the model didn't predict it).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$Recall = \frac{TP}{TP + FN}$$

Now as we notice, we can see that the Numerator in both Precision and Recall is the same: True Positives. But in the denominator, Precision counts the False Positives while Recall takes False Negatives into consideration.

Let us ponder… Which one do you think is better? Precision or Recall? Why?

_____
_____
_____
_____
_____
_____

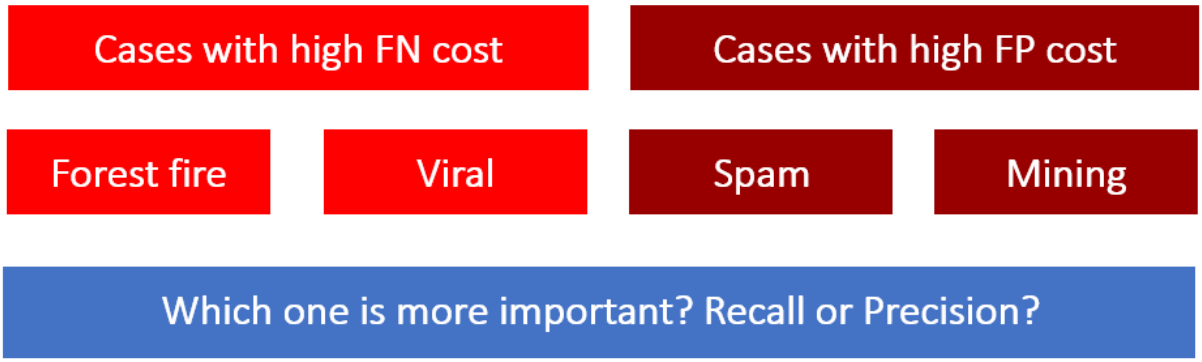
Which Metric is Important?

Choosing between Precision and Recall depends on the condition in which the model has been deployed. In a case like Forest Fire, a False Negative can cost us a lot and is risky too. Imagine no alert being given even when there is a Forest Fire. The whole forest might burn down.

Another case where a False Negative can be dangerous is Viral Outbreak. Imagine a deadly virus has started spreading and the model which is supposed to predict a viral outbreak does not detect it. The virus might spread widely and infect a lot of people.

On the other hand, there can be cases in which the False Positive condition costs us more than False Negatives. One such case is Mining. Imagine a model telling you that there exists treasure at a point and you keep on digging there but it turns out that it is a false alarm. Here, False Positive case (predicting there is treasure but there is no treasure) can be very costly.

Similarly, let's consider a model that predicts that a mail is spam or not. If the model always predicts that the mail is spam, people would not look at it and eventually might lose important information. Here also False Positive condition (Predicting the mail as spam while the mail is not spam) would have a high cost.



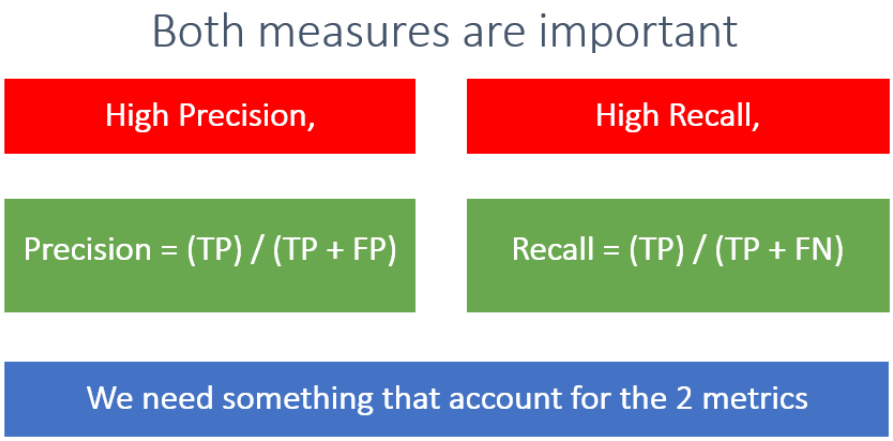
Think of some more examples having:

- High False Negative cost

_____
_____
_____
_____

- High False Positive cost

_____
_____
_____
_____



To conclude the argument, we must say that if we want to know if our model's performance is good, we need these two measures: Recall and Precision. For some cases, you might have a High Precision but Low Recall or Low Precision but High Recall. But since both the measures are important, there is a need of a parameter which takes both Precision and Recall into account.

F1 Score

F1 score can be defined as the measure of balance between precision and recall.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Take a look at the formula and think of when can we get a perfect F1 score?

An ideal situation would be when we have a value of 1 (that is 100%) for both Precision and Recall. In that case, the F1 score would also be an ideal 1 (100%). It is known as the perfect value for F1 Score. As the values of both Precision and Recall ranges from 0 to 1, the F1 score also ranges from 0 to 1.

Let us explore the variations we can have in the F1 Score:

Precision	Recall	F1 Score
Low	Low	Low
Low	High	Low
High	Low	Low
High	High	High

In conclusion, we can say that a model has good performance if the F1 Score for that model is high.

Let's practice!

Let us understand the evaluation parameters with the help of examples.

Challenge

Find out Accuracy, Precision, Recall and F1 Score for the given problems.

Scenario 1:

In schools, a lot of times it happens that there is no water to drink. At a few places, cases of water shortage in schools are very common and prominent. Hence, an AI model is designed to predict if there is going to be a water shortage in the school in the near future or not. The confusion matrix for the same is:

The Confusion Matrix	Reality: 1	Reality: 0
Predicted: 1	22	12
Predicted: 0	47	118

Scenario 2:

Nowadays, the problem of floods has worsened in some parts of the country. Not only does it damage the whole place but it also forces people to move out of their homes and relocate. To address this issue, an AI model has been created which can predict if there is a chance of floods or not. The confusion matrix for the same is:

The Confusion Matrix	Actual: 1	Actual: 0
Predicted: 1	0	3
Predicted: 0	3	94

Scenario 3:

A lot of times people face the problem of sudden downpour. People wash clothes and put them out to dry but due to unexpected rain, their work gets wasted. Thus, an AI model has been created which predicts if there will be rain or not. The confusion matrix for the same is:

The Confusion Matrix	Actual: 1	Actual: 0
Predicted: 1	5	0
Predicted: 0	45	50

Scenario 4:

Traffic Jams have become a common part of our lives nowadays. Living in an urban area means you have to face traffic each and every time you get out on the road. Mostly, school students opt for buses to go to school. Many times the bus gets late due to such jams and students are not able to reach their school on time. Thus, an AI model is created to predict explicitly if there would be a traffic jam on their way to school or not. The confusion matrix for the same is:

The Confusion Matrix	Actual: 1	Actual: 0
Predicted: 1	50	50
Predicted: 0	0	0