

Receiver operator characteristic curve(ROC)

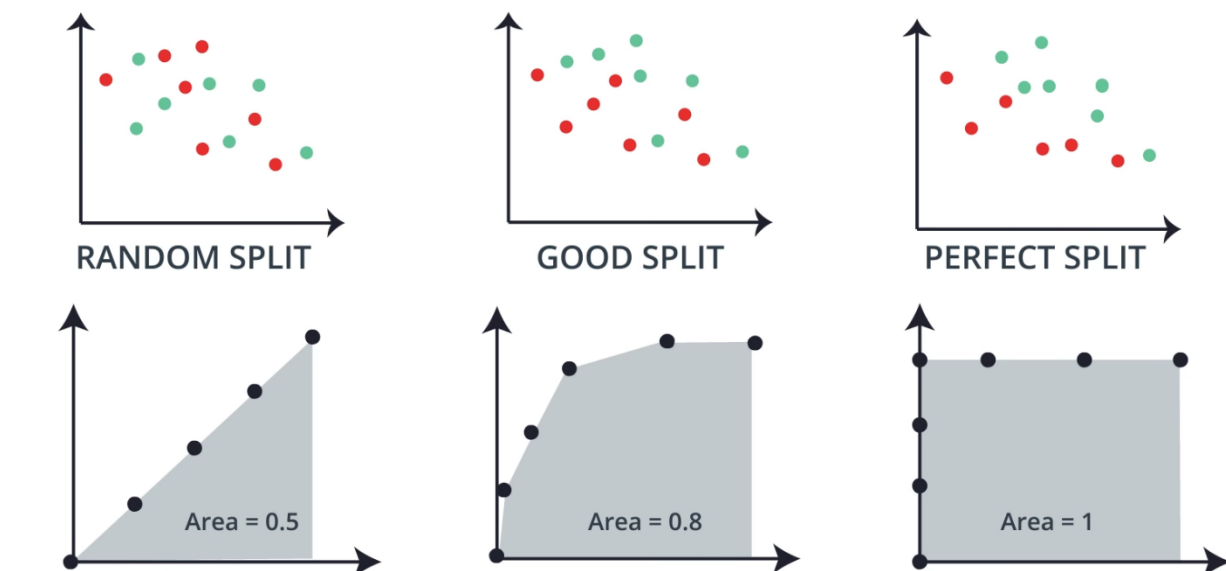
Consider this data which is now one dimensional, so all the red and blue points lie in one line and we want to find the correct split.

- So, we can have a split around here or maybe here or here, all of them are good splits. This is a **good split**.
- Now we can look at this data, which as you can see is perfectly separable over here. This is a **perfect split**.
- Finally, we have this data over here which is pretty much random and there's not much to split here. It seemed that anywhere we put the boundary, we'll have about half blue, half red points on each side. This is a **bad split** or a **random split**.

Now what we want is to come up with a metric or some number that is high for the perfect split, medium for the good split, and low for the random split. In fact, something that gives the perfect split a score of 1.0, the good split something around 0.8, and the random split something around 0.5.

By finding different thresholds for our classification metrics, we can measure the area under the curve (where the curve is known as a ROC curve). Similar to other metrics, when the AUC is higher (closer to 1), this suggests that our model performance is better than when our metric is close to 0.

◦ AREA UNDER A ROC CURVE



Let's see how to construct these numbers. Let's take our good data and cut it. We'll calculate two ratios:

- The first one is a true positive rate, which means out of all the positively labeled points, how many did we classify correctly? That means the number of true positives divided by the total number of positively labeled points. So let's see how much this is. There are 7 positively labeled numbers and 6 of them have been correctly labeled positive, so this ratio is $6/7$ or 0. 857.
- Now let's look at the false positive rate, which means out of all the negative points, how many of them did the model incorrectly think they were positives? So out of the 7 negatively labeled points, the model thought 2 of them were positive. So the false positive rate is $2/7$ or 0. 286.

We'll just remember these two numbers. Now what we'll do is we'll move the boundary around and calculate the same pair of numbers.

- What is the true positive rate over here? Well, the model thinks everything is positive. So in particular, all the positives are true positives. So the true positive rate is $7/7$, which is 1 .
- For the false positive rate, well, since the model thinks everything is positive, then all the negatives are false positives. So the false positive rate is again $7/7$, which is 1 .

So again, we'll remember these two values, one and one.

Let's go to the other extreme. Let's put the bar over here

- Let's see what the true positive rate is. Well, the model thinks nothing is positive so in particular, there are no true positives and the ratio is $0/7$, which is 0.
- For the false positive rate, well, again, the model thinks nothing is positive, so there are no false positives and the ratio is $0/7$, which again is 0 .

We'll remember these two numbers.

We can see that no matter how the data looks, the two extremes will always be 1, 1 and 0, 0.

Now, we can do this for every possible split and record those numbers. So here we have a few of them that we've calculated.

Now, the magic happens.

- We plot these numbers in the plane and we get a curve. We calculate the area under the curve and here we get around 0.8.
- Next, let's do the same thing for the perfect split. Here are all the ratios. Notice that if the boundary is on the red side, then the true positive ratio is one since every positive number has been predicted positive. Similarly, if the boundary is on the blue side, then every negative number has been predicted negative and so the false positive ratio is 0. In particular, at the perfect split point, we have a 0, 1. Thus, when we plot these

numbers, the curve looks like a square and the square has area, one, which means the area under the ROC curve for the perfect split is 1.

- Finally, we do this for the random split. Here you can try it on your own, but basically since every split leaves on each side around half blue, half red points, then each pair of numbers will be close to each other, and the curve will be very close to being just a diagonal between zero, zero and one, one. So if the model is random, then the area under the ROC curve is around 0.5.

Summary

We have three possible scenarios; some random data which is hard to split, some pretty good data which we can split well making some errors, and some perfectly divided data which we can split with no errors. Each one is associated with a curve. The areas under the curve are close to 0.5 for the random model, somewhere close to one for the good model, so around 0.8, and one for the perfect model.

The closer your area under the ROC curve is to one, the better your model is. Can the area under the curve be less than 0.5? In fact, yes. It can be all the way to zero. How would a model look if the area under the curve is zero? Well, it will look more backwards. It'll have more blue points in the red area and the red points in the blue area, so maybe flipping the data may help.

Quiz Question

True or False: The closer your area under the ROC curve is to zero , the better your model is.

- a. True
- b. False

Quiz Question

Which of the following statements are true about the area under the curve? (There may be more than one correct answer)

- a. around 0.5 for the random model
- b. 1.0 for the perfect model.
- c. Zero for the perfect model.
- d. around 0.8, and one for the good model.

F1 score

Let's introduce something a bit more general. If we have precision and recall, then the F1 score is somewhere around the middle. That gives us much importance to both. Of course, if one of them is smaller, it raises a flag, but it treats them both the same. Now let's say our model cares a bit more about precision than recall, then we want something more skewed towards precision.

So, we'll say it's F0.5 score. So we call that beta. Beta is 0.5.

The smaller the beta, the more towards precision that we get.

Or if we want our model to care more about the recall, then we pick a larger beta.

The larger the beta, the more towards recall that we get.

Fraud detection example

In the fraud detection example, which beta should we use? Since it needs to be a high recall model since we need to catch all the fraud cases, and it's okay if we accidentally detect and investigate some that are not.

So something like F10. But then maybe, we're sacrificing too much precision, and we're accidentally sending our customers too many notifications about their transactions without them being fraudulent, and they're starting to get annoyed. So, we can move a bit toward say F2. But then maybe we discover that we need to focus a bit more on recall because we really don't want to miss too many fraudulent transactions, so we go here to F5.

It's not an exact science. Finding a good value of beta requires a lot of intuition of your data and a lot of experimentation.

Quiz

Now, let's test our knowledge. So, let's look at three possible models:

- In the first one, we are NASA, and we have a model for detecting malfunctioning parts on a spaceship.
- In the second, we have a video recommender system, and we have a model for sending users phone notifications about new videos they may like.
- And in the third one, we are a store, and we have a model for sending free samples in the mail to potential clients.

And let's say one of them has an F beta score of F1, the other one is F0.5, and the other one is F2. Which one is which?

◦ QUIZ: F_β SCORE

- ☐ DETECTING MALFUNCTIONING PARTS ON A SPACESHIP
- ☐ SENDING PHONE NOTIFICATIONS ABOUT VIDEOS A USER MAY LIKE
- ☐ SENDING FREE SAMPLES IN THE MAIL TO POTENTIAL CLIENTS



PRECISION

$F_{0.5}$ SCORE

F_1 SCORE

F_2 SCORE



RECALL

Quiz Question

Out of the following three models, which one should have an F-beta score of 2, 1, and 0.5?

Match each model with its corresponding score.

- Detecting malfunctioning parts in a spaceship
- Sending phone notifications about videos a user may like
- Sending promotional material in the mail to potential clients

Model

Spaceship

Notifications

Promotional Material

F-beta Score

0.5_____

2_____

1_____

Averaging precision and recall

- The medical model has a precision of 55.7 % and a recall of 83.3%. It's supposed to be a high-recall model.
- The spam detector has a precision of 76.9 % and a recall of 37%. It's supposed to be a high-precision model.

Do we want to be carrying two numbers around all the time? We kind of want to only have one score. So the question is, how do we combine these two scores into one?

One answer is simply to take the average. Let's take the average of precision and recall.

- On the left, we get 69.5 percent.
- On the right, we get 56.95 percent.

And that's an okay metric, but not very different from accuracy. The way to see how this average is **not** the best idea is to try it in the extreme example.

Credit card fraud example 1

Precision

We have a bunch of good and fraudulent credit card transactions. Let's pick a terrible model one that says, "All transactions are good." What is the precision of this model? Well, the precision is, "out of the ones we classified as bad, how many of them are bad?" That's a question about numbers because we didn't label anything as fraudulent. So it's kind of zero divided by zero, which is undefined. But it makes sense to think of it as 100 percent since we made zero mistakes among the ones who predicted positive, which is what precision tries to measure. So let's say this model has 100 percent precision.

Recall

The recall is, "how many of the fraudulent transactions did we catch?" Well, since we caught none, this number is zero.

The average

The average between precision and recall is 50 percent since it's the average of 100 and zero. Now the question is, do I want to give this horrendous model of 50 percent? It seems like a really high score for such a lousy model. I kind of want to give it a much lower score, perhaps even zero.

Credit card fraud example 2

Precision

Now let's try the opposite. And let's try the model that says, "all transactions are fraudulent. " What is the precision of this model? Out of all the transactions that I said are fraudulent, 472 were actually fraudulent. The precision is $472 / 284,807$, which is 0.16%.

Recall

The Recall is actually pretty good because out of the 472 fraudulent transactions, I caught all of them. So the recall is $472 / 472$, which is 100%.

The average

The average of the two is the average of 0.16 and 100, which is 50.08%, a very high score for a really lousy model. So we want to give it a lower score or maybe something close to zero.

Averaging conclusion

Averaging is **not** the greatest thing in principle if either precision or recall is very low. We want the number to be low even if the other one is high.

Harmonic mean

There's another type of average called the harmonic mean, and it works as follows. Imagine we have two numbers, X and Y, with $X < Y$. And we have their arithmetic mean, which is the average, $(X+Y)/2$. And we have something called the harmonic mean which is defined by

$$2xy / x+y$$

It's kind of an average of the two in the sense that if the two numbers are equal, we get X or Y, and it always lies between X and Y.

It's a mathematical fact that the harmonic mean is always less than the arithmetic mean. So it's closer to the smaller number than to the higher number.

F1 score

If the precision is one and the recall is zero, the average is 0.5, but the harmonic mean is, if we plug in the formula, zero. Another example, if the precision is 0.2 and the recall is 0.8, then the arithmetic mean is 0.5, but the harmonic mean is 0.32. So it's closer to the lower number. From now on, we will not be using the average or arithmetic mean, but we'll be using the harmonic mean, and that's going to be called the **F1 score**, which is closer to the smallest value between precision and recall. If one of them is particularly low, the F1 score can raise a flag.

$$F_{\beta} = (1 + \beta^2) * \text{Precision} \cdot \text{Recall} / (\beta^2 * \text{Precision} + \text{Recall})$$

You can see that the β (beta) parameter controls the degree to which precision is weighed into the F score, which allows precision and recall to be considered simultaneously. The most common value for beta is 1, as this is where you are finding the harmonic average between precision and recall.

Quiz Question

Which of the following is not true about an F1 score?

- a. It is the harmonic mean of the dataset
- b. It will always be closer to the smaller result of precision and recall values
- c. It will always be closer to the larger result of precision and recall values

Credit card fraud example

Go back to the credit card fraud example and calculate the F1 score. Since it's going to be the harmonic mean between the precision, which is 100%, and the recall, which is 0, we can plug in the formula and actually get an F1 score of zero. This is much closer to what the models should score.

So, in the exercise below, we'll let you calculate the F1 score of the medical model and the spam email model.

F1 Score Quiz

For the following, remember that the formula for F1 Score is:

$$\text{F1 Score} = 2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall})$$

Precision Quiz

If the Precision of the medical model is **55.6%**, and the Recall is **83.3%**, what is the F1 Score? (Please write your answer as a percentage, and round it to 1 decimal point.)

Answer: _____46.64_____

Recall

Recall focuses on the **actual** "positive" values in your dataset. By optimizing based on recall values, you are determining if you are doing a good job of predicting the positive values **without** regard to how you are doing on the **actual** negative values. If you want to perform something similar to recall the **actual** 'negative' values, this is called specificity ($TN / (TN + FP)$).

	Spam (positive)	Not Spam (negative)
Spam (positive)	True Positive(TP)	False Positive(FP)
Not Spam (negative)	False Negative(FN)	True Negative(TN)

Medical model

Let's look at the second metric called **recall**. Recall is the answer to the following question, *"out of the points that are labeled positive, how many of them were correctly predicted as positive?"* So in the medical model, a recall is the answer to the following question, *"out of the sick patients, how many did we correctly diagnose as sick?"* Remember:

- **Precision** was, *"out of the patients were diagnosed as sick, how many were actually sick?"*
- **Recall** is the opposite, *"out of the patients that are sick, how many did we correctly diagnose as sick?"*

It can be seen as a reach of the algorithm. How many of the positive points did I manage to catch? As we can see, this row catches a critical case labeled by an X, so we can see that recall is important in the medical model. Now to calculate recall, we do the following; from the 1,200 sick patients, how many did we diagnose correctly? That's 1,000 divided by 1,200, which is 83.3 percent. This model better have a higher recall because we're trying to correctly catch as many of the sick people as possible.

Spam email model

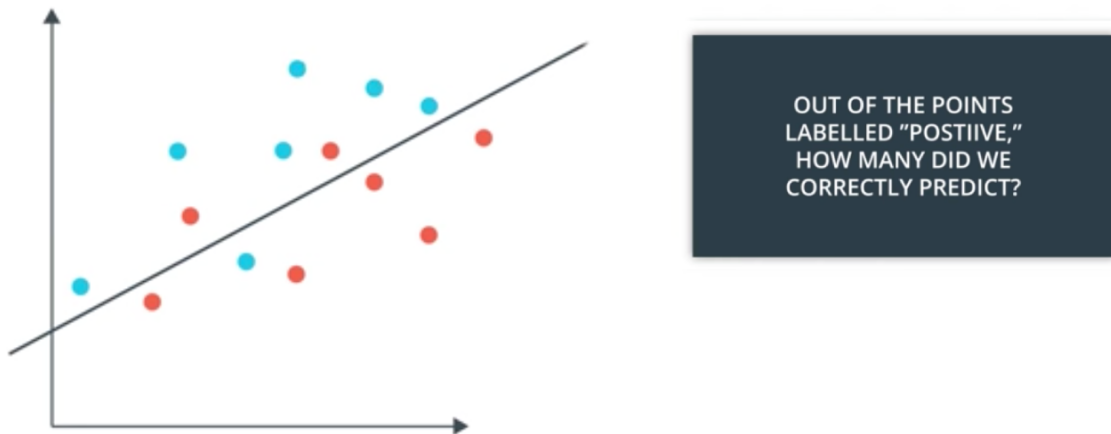
Recall is calculated as follows; from the spam emails, how many of them do we correctly send to the spam folder? You can see the row with 100 correctly sent to the spam folder divided by 270 total spam emails. It's a low number. It's 37 percent. But remember, we are worried about

avoiding this X, since we don't mind if we don't catch all the spam emails as long as the ones we caught are spam.

It's okay that this model has a low recall.

Action Required

Let's do a similar exercise as before. In this linear model over here, what is the recall?



In this image, the blue points are labelled positive, and the red points are labelled negative. Furthermore, the points on top of the line are predicted to be positive, and the points below the line are predicted to be negative. Note: This image has one extra point than the one in the video. This is the correct image.

Recall Quiz

What is the recall of the linear model above? Please write your number as a decimal, like **0.45** or as a fraction, like **3/5**.

Answer: 6/14*100=42.85

Quiz Question

Recall is the answer to the following question:

- "Out of the points that are labeled positive, how many of them were correctly predicted as negative?"

- b. "Out of the points that are labeled positive, how many of them were correctly predicted as positive?"

Precision

Precision focuses on the **predicted** "positive" values in your dataset. By optimizing based on precision values, you are determining if you are doing a good job of predicting the positive values, as compared to predicting negative values as positive.

	Spam (positive)	Not Spam (negative)
Spam (positive)	True Positive(TP)	False Positive(FP)
Not Spam (negative)	False Negative(FN)	True Negative(TN)

Medical model

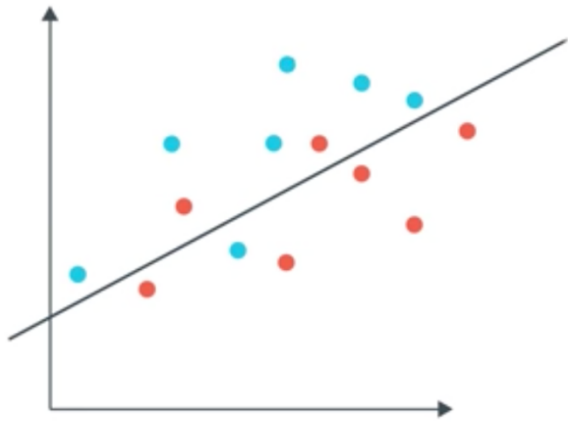
So let's define the precision metric as follows. Here's the confusion matrix of the medical model, and we've added a red X in the spot that we really want to avoid, which is the false negatives. precision is the answer to the question, "*Out of all the points predicted to be positive, how many of them were actually positive?*" In this case, the question translates to, out of all the patients that we diagnosed as sick, how many were actually sick? So precision is this column because this column is the sick patients that we diagnose as sick. So it is 1,000 that were correct, divided by 1,800, which is the total number of patients diagnosed as sick. This number is 55.7 percent. It's not a high number because this is not a very precise model. But, again, this is okay because what we're doing is avoiding this red X.

Spam email model

What is this precision? Now, we know that in this model, precision is very important because the red X that we're avoiding is in this column. The red X is the non-spam email that was accidentally sent to the spam folder. So those 30 errors are really bad and we want to avoid them. So, again, precision says, "*Out of all the emails sent to the spam folder, how many of them were actually spam?*" So we have 100 which are correct, divided by 130 which is all the ones we've sent to the spam folder. This number is 76.9 percent, which is higher. This is better since this model needs high precision, so the number better be big.

Action Required

Now, let's do an exercise. Let's go to a linear model over here. What is the precision of this linear model?



OUT OF THE POINTS WE HAVE
PREDICTED TO BE POSITIVE,
HOW MANY ARE CORRECT?

In this image, the blue points are labelled positive, and the red points are labelled negative. Furthermore, the points on top of the line are predicted to be positive, and the points below the line are predicted to be negative.

Precision Quiz

What is the precision of the linear model above? Please write the number as a decimal, like **0.45** or as a fraction, like **3/5**.

Answer: 6/8=0.75

Quiz Question

So precision will be the answer to the question?

- "Out of all the points predicted to be positive, how many of them were actually positive?"
- "Out of all the points predicted to be positive, how many of them were actually negative?"






Medical model

Recall that for the medical model, we have four possibilities:

- True positive when the patient is sick and we correctly diagnosed him as sick
- True negative when the patient is not sick and we correctly diagnose them as not sick
- False positive when the patient is healthy and we incorrectly diagnose them as sick
- False negative when the patient is sick and we incorrectly diagnose them as healthy

Quiz 1

Which one do you think is worse, a false positive or a false negative? In other words, what is the worst mistake, to misdiagnose a healthy patient as sick or a sick patient as healthy? Respond to the exercise below.

		Diagnosis	
Patients		DIAGNOSED SICK	DIAGNOSED HEALTHY
	SICK		  FALSE NEGATIVE
	HEALTHY	  FALSE POSITIVE	

Quiz Question

In the medical example, what is worse, a False Positive, or a False Negative?

- False Positive
- False Negative




Spam email model

Imagine we have two emails: one non-spam from our grandma telling us she baked cookies and one spam from a questionable source. Recall that for this model, we again have four possibilities:

- True positive when the spam e-mail gets correctly sent into the spam folder
- True negative when our grandma's e-mail gets correctly sent to the inbox folder
- False positive when grandma's email accidentally gets sent to the spam folder
- False negative when a spam email accidentally gets sent to our inbox

Quiz 2

Which one do you think is worse, a false positive or a false negative? In other words, what is a worse mistake, to accidentally send your grandma's email to the spam folder or to accidentally send the spam email into your inbox?

		Folder	
Emails		SENT TO SPAM	SENT TO INBOX
	SPAM		 FALSE NEGATIVE
	NOT SPAM	 FALSE POSITIVE	

Quiz Question

In the spam detector example, what is worse, a False Positive, or a False Negative?

- False Positive
- False Negative

Accuracy may not always be the best metric to use

Detecting credit card fraud

We have a bunch of data in the form of credit card transactions. There are 284,335 good transactions and 472 fraudulent transactions. This is actually real data. So, let's try to come up with a model that has great accuracy..in other words, a model that is correct most of the time. Can you think of a model that has over 99 percent accuracy?

All the transactions are good

Let's use a model that says, all the transactions are good. This model has over 99 percent accuracy, which means it is correct over 99 percent of the time. Now, what is the accuracy of this model, namely how many times are we correct? Well, we're correct 284,335 times out of 384,887 times. And that is 99.83 percent. Logically, this model must be pretty good if it's accurate is that high, right? No! This model is not catching any of the bad ones. And the point of the model is to catch the fraudulent transactions.

All transactions fraudulent

Can we get a model that catches all the bad transactions? This model catches all fraudulent transactions. Is that a good model? No! That's also a terrible model since it's accidentally catching all the good ones.

It's pretty tricky to just look at accuracy and use that to evaluate our model because it may completely miss the point when the data is skewed like this one.

Quiz Question

Accuracy is the answer to the question

- a. "Out of all the transactions, how many did we classify correctly?"
- b. "Out of all the good transactions, how many did we classify correctly?"
- c. "Out of all the fraudulent transactions, how many did we classify correctly?"

How accurate is your model?

Accuracy is the answer to the question, "Out of all the patients, how many did we classify correctly?" The answer is the ratio between the number of correctly classified points and the number of total points. In this example, we can see that we have correctly classified 9,000 patients, which is a sum of 1,000 healthy and 8,000 sequences. So, the **accuracy** is this number divided by the total number of patients, which is 10,000. This gives us an accuracy of 90%.

Accuracy can easily be calculated as you can learn by using the accuracy score function.

	Spam (positive)	Not Spam (negative)
Spam (positive)	True Positive(TP)	False Positive(FP)
Not Spam (negative)	False Negative(FN)	True Negative(TN)

Often accuracy is not the only metric you should be optimizing on. This is especially the case when you have a class imbalance in your data. Optimizing on only accuracy can be misleading in how well your model is truly performing. With that in mind, you saw some additional metrics.

Example: how much is the accuracy of the spam detector model?

This is the answer to the question, "Out of all the emails, how many did we classify correctly?" Since we have correctly classified 800 out of the 1,000 emails, then the answer is 80%.

Quiz Question

Accuracy of 100% on a training set is like ____.

- a.ignoring the training data. A model designed to have 100% accuracy on training data is likely to generalize well to new data.
- b.memorizing the training data. A model designed to have 100% accuracy on training data is unlikely to generalize well to new data.

- c.memorizing the training data. A model designed to have 100% accuracy on training data is likely to generalize well to new data.

Accuracy Quiz

What is the accuracy of the model above? Please enter the answer as a percentage, with two decimals. For example, 54. 75.

Answer: $(11/14)*100=78.57$

How good is the model?

This is a difficult question. But in this section, we'll learn a few different metrics that will tell us how good our model is. So we're going to look at two main examples.

Example 1

The first example is a model that will help us detect a particular illness, and tell if a patient is healthy or sick. There are four possible cases:

- When a patient is sick, the model correctly diagnoses them as sick. This is a sick patient, I will send them for further examination or for treatment. This case, we'll call it a **true positive**.
- When a patient is healthy and the model correctly diagnosed him as healthy, this is a healthy patient that we'll send home. This case, we call it a **true negative**.
- When a patient is sick and the modeling correctly diagnoses them as healthy. This is a mistake, and it means we'll be sending a sick patient back home with no treatment. This is called a **false negative**.
- And finally, when a patient is healthy and the model incorrectly diagnoses them as sick. This is also a mistake, and it means we'll be sending a healthy person for further examination or treatment. This is called a **false positive**.

Confusion matrix

This is a table that will describe the performance of a model. In this model, we have 10,000 patients. A thousand of them are sick and have been correctly diagnosed as sick. We call these true positives. 200 of them are sick and have been incorrectly diagnosed as healthy. So we call them false negatives. 800 patients are healthy and have been incorrectly diagnosed as sick. We call these false positives. And finally, 8,000 patients are healthy and have been correctly diagnosed as healthy. We call these true negatives. The confusion matrix is a simple table that stores these four values.

Example 2

The second example will be a spam detector, which will help us determine if an email is spam or not. There are four possible cases:

- When we get a spam email and the classifier sends it to a spam folder correctly, which is a **true positive**.

- When we get a spam email and the classifier incorrectly sends it to our inbox, this is a **false negative**.
- When we get a good email, for example, from our grandma and the classifier incorrectly sends it to our spam folder, this is called a **false positive**.
- And finally, when we get a good email the classifier correctly sends it to our inbox, which is a **true negative**.

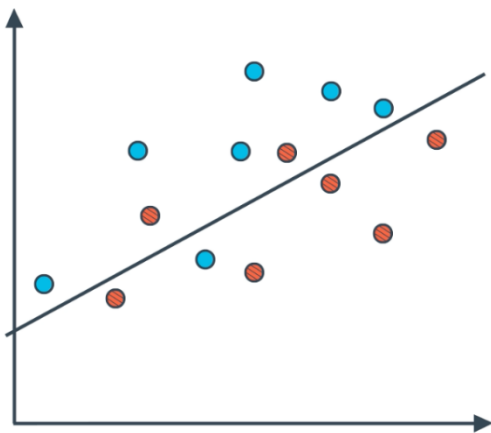
Confusion matrix

We have a pool of a thousand emails. Out of these emails, 100 spam emails have been correctly sent to the spam folder. 170 spam emails have been incorrectly sent to the inbox. 30 non-spam emails have been incorrectly sent to the spam folder. And finally, 700 non-spam emails have been correctly sent to the inbox.

Action Required

Now it's your turn to create a confusion matrix. Look at this data where the blue points are positive, and the red points are negative. The model we've trained is the line that separates them, with the positive region being at the top and the negative region at the bottom. Now figure out the confusion matrix for the number of true positives, true negatives, false positives, and false negatives.

○ CONFUSION MATRIX



	Guessed Positive	Guessed Negative
Positive	True Positives	False Negatives
Negative	False Positives	True Negatives

In this image, the blue points are labelled positive, and the red points are labelled negative. Furthermore, the points on top of the line are predicted (guessed) to be positive, and the points below the line are predicted to be negative.

Confusion Matrix Quiz

How many True Positives, True Negatives, False Positives, and False Negatives, are in the model above? Please enter your answer in that order, as four numbers separated by a comma and a space. For example, if your answers are 1, 2, 3, and 4, enter the string 1, 2, 3, 4. Remember, in the image above the blue points are considered positives and the red points are considered negatives.

Answer: _____7,1,2,5_____

Type 1 and Type 2 Errors

Sometimes in the literature, you'll see False Positives and False Negatives as Type 1 and Type 2 errors. Here is the correspondence:

- **Type 1 Error (Error of the first kind, or False Positive):** In the medical example, this is when we misdiagnose a healthy patient as sick.
- **Type 2 Error (Error of the second kind, or False Negative):** In the medical example, this is when we misdiagnose a sick patient as healthy.

Mean-Squared Error (MSE) Evaluating regression models

Mean Absolute Error (MAE)

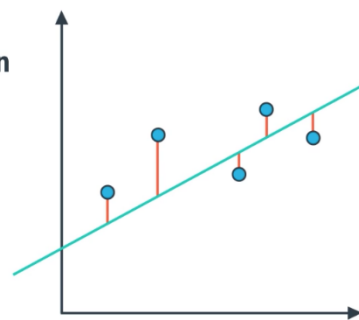
- In this example, we have some points and we have drawn a line that fits these points. Now we would like to check how bad this line is. A good way to do this is to just add the absolute values of the distances from the points to the line. Mean absolute error is very easy to calculate in SKLearn. Create a linear regression classifier object called `classifier`.
- Then we use the function to fit the line. Now we can refer to the points as `y` and to the predictions made by the model as `classifier.predict(X)`. We will call these guesses. Finally, the mean absolute error is just calculated with the mean absolute error function. The mean absolute error has a problem which is that the absolute value function is not differentiable. This may not be good if we want to use methods such as gradient descent.
- This is a useful metric to optimize when the value you are trying to predict follows a skewed distribution. Optimizing on an absolute value is particularly helpful in these cases because outliers will not influence models attempting to optimize on this metric as much as if you use the mean squared error. The optimal value for this technique is the median value.

```
from sklearn.metrics import mean_absolute_error
from sklearn.linear_model import LinearRegression

classifier = LinearRegression()
classifier.fit(X,y)

guesses = classifier.predict(X)

error = mean_absolute_error(y, guesses)
```



Mean-Squared Error (MSE)

The mean squared error is by far the most used metric for optimization in regression problems. Similar to MAE, you want to find a model that minimizes this value. This metric can be greatly impacted by skewed distributions and outliers. When a model is considered optimal via MAE,

but not for MSE, it is useful to keep this in mind. In many cases, it is easier to actually optimize on MSE, as the quadratic term is differentiable. However, an absolute value is not differentiable. This factor makes this metric better for gradient-based optimization algorithms.

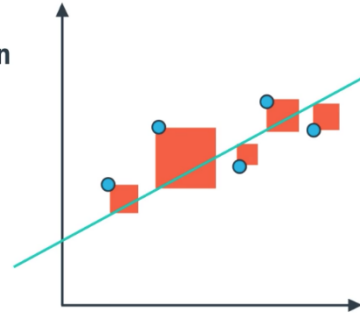
For this metric, we add the squares of the distances between the points and the line. Mean square error is also very easy to calculate in SKLearn, except now we will use the mean square error function.

```
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression

classifier = LinearRegression()
classifier.fit(X,y)

guesses = classifier.predict(X)

error = mean_squared_error(y, guesses)
```



Mean Squared Error

R2 score

- Finally, the r^2 value is another common metric when looking at regression values. Optimizing a model to have the lowest MSE will also optimize a model to have the highest R^2 value. This is a convenient feature of this metric. The R^2 value is frequently interpreted as the 'amount of variability captured by a model'. Therefore, you can think of MSE, as the average amount you miss across all the points and the R^2 value as the amount of the variability in the points that you capture with a model.
- R^2 score is based on comparing our model to the simplest possible model. Let's think, what is the simplest possible model that fits a bunch of points? Well, a pretty simple one is just to take the average of all the values and draw a horizontal line through them, and we can calculate the mean squared error for this model. We would hope that the mean squared error for the simple model is larger than the error for a linear regression model. The question is, how much larger? Well, we can divide the error for the linear regression model by the error for the simple model, and then subtract the result from 1, and we will call these the R^2 score. If the model is not very good, then the two errors should be similar and this quantity here should be close to one. The whole R^2 score should be close to zero. If the model is good, then the mean squared error for the linear regression model should be a lot smaller than the mean square for the simple model. Therefore, this ratio should be small and then the R^2 squared should be very close to one.
- As you can see in the image below, if the R^2 score is close to one, then the model is good. If it's close to zero, then the model is not much better than just guessing the

average of the values of the points R2 score is very simple to calculate in SKLearn with the R2 score function.

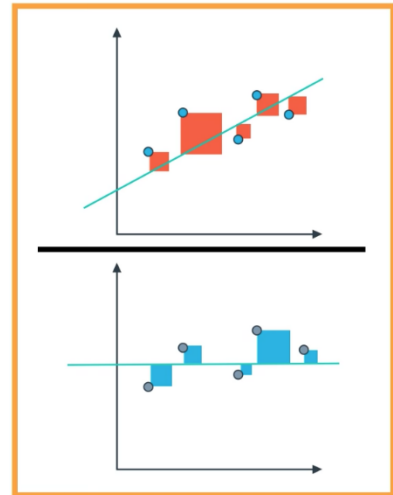
BAD MODEL

The errors should be similar.
R2 score should be close to 0.

GOOD MODEL

The mean squared error for the linear regression model should be a lot smaller than the mean squared error for the simple model.
R2 score should be close to 1.

$$R^2 = 1 -$$



R2 Score

Summary

Do you want to measure how well your algorithms are performing on predicting numeric values? In these cases, there are three main metrics that are frequently used. **mean absolute error**, **mean squared error** and **r2** values.

As an important note, optimizing on the mean absolute error may lead to a different 'best model' than if you optimize on the mean squared error. However, optimizing on the mean squared error will **always** lead to the same 'best' model as if you were to optimize on the **r2** value.

Again, if you choose a model with the best r2 value (the highest), it will also be the model that has the lowest (MSE). Choosing one versus another is based on which one you feel most comfortable explaining to someone else.

Quiz Question

Connect the evaluation metric with the correct description

Divide the error for the linear regression model by the error for the simple model, and then subtract the result from 1

Answer: _____

Add the squares of the distances between the points and the line

Answer: _____

Add the absolute values of the distances from the points to the line.

Answer: _____

Extra Resource: <https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks>