Resampling Methods and Evaluation

A unit of Machine Learning

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Contents

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation

- ★ How do we know if our model is functional? If we have trained it well?
- ★ All of this can be determined by seeing how our model performs on previously unseen data, data that is completely new to it.
- ★ We need to ensure that the accuracy of our model remains constant throughout. In other words, we need to **validate our model.**
- ★ Using **cross-validation** in machine learning, we can determine how our model is performing on previously unseen data and test its accuracy.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Why models lose stability?

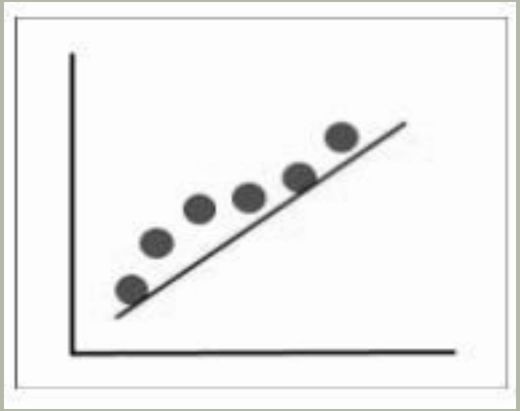
- ★ **Stability :** Any machine learning model needs to consistently predict the correct output across a variation of different input values, present in different datasets. This characteristic of a machine learning model is called stability.
- ★ A model can lose stability in two ways:

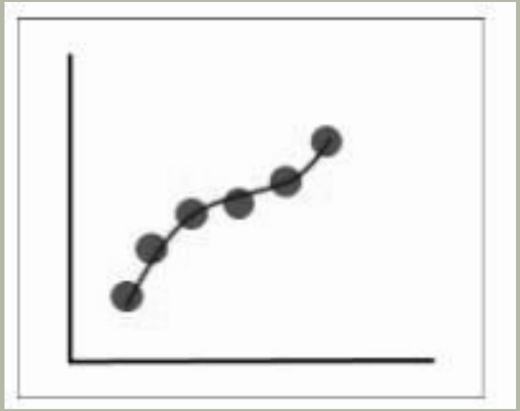
  1. **Underfitting:** It occurs when the model does not fit properly to training data. It does not find patterns in the data and hence when it is given new data to predict, it cannot find patterns in it too. It under-performs on both known and unseen data.

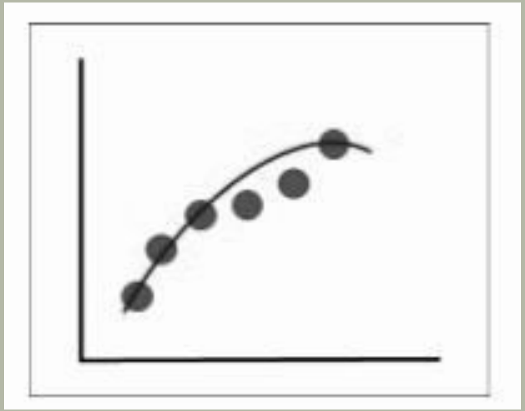
  2. **Overfitting:** When the model trains well on training data and generalizes to it, but fails to perform on new, unseen data. It captures every little variation in training data and cannot perform on data that does not have the same variations.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Why models lose stability?







Underfitting

Overfitting

Optimal Model

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Why models lose stability?

- ★ In Figure 1, we can see that the model does not quite capture all the features of our data and leaves out some important data points. This model has generalized our data too much and is under fitted.
- ★ In Figure 2, our model has captured every single aspect of the data, including the noise. If we were to give it a different dataset, it would not be able to predict it as it is too specific to our training data, hence it is overfitted.
- ★ In figure 3, the model captures the intricacies of our model while ignoring the noise, this model is our optimal model.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation

- ★ **Background :**While choosing machine learning models, we need to compare models to see how different models perform on our dataset and to choose the best model for our data. However, data is usually limited, our dataset might not have enough data points or may even have missing or wrong data.

  Further, if we have fewer data, training, and testing on the same portion of data does not give us an accurate view of how our model performs. Training a model on the same data means that the model will eventually learn well for only that data and fail on new data, this is called overfitting. This is where cross-validation comes into the picture.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

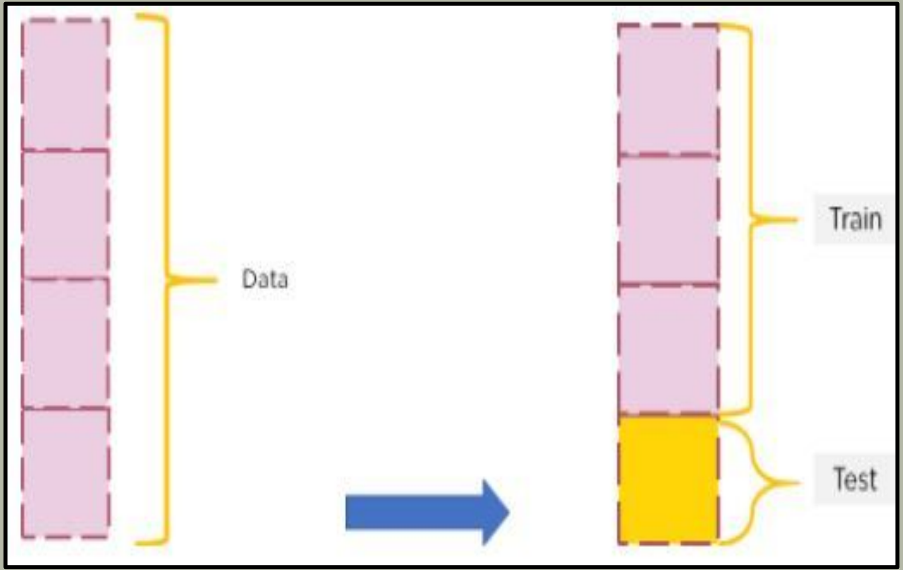
Cross validation

★ **Introduction:** Cross-Validation in machine learning is a technique that is used to train and evaluate our model on a portion of our database, before re-portioning our dataset and evaluating it on the new portions.

This means that instead of splitting our dataset into two parts, one to train on and another to test on, we split our dataset into multiple portions, train on some of these and use the rest to test on. We then use a different portion to train and test our model on. This ensures that our model is training and testing on new data at every new step.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation - Idea

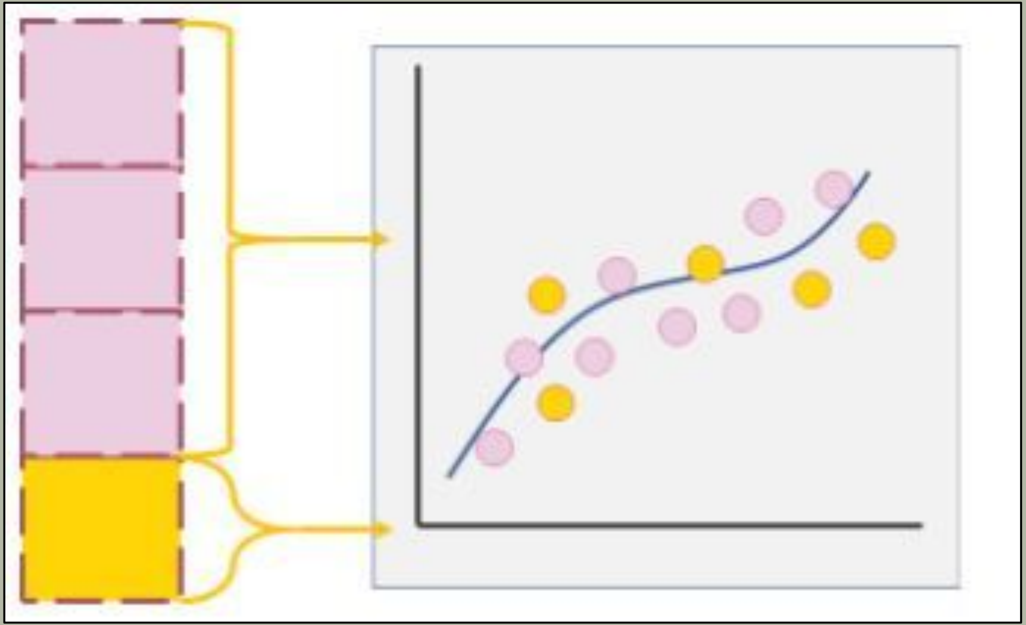


Partitioning dataset for cross-validation

Consider the block below to represent the entirety of our data. We partition the dataset into training and testing data. The training data will be used by our model to learn. The testing dataset will be used by our model to predict unseen data. It is used to evaluate our model's performance.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

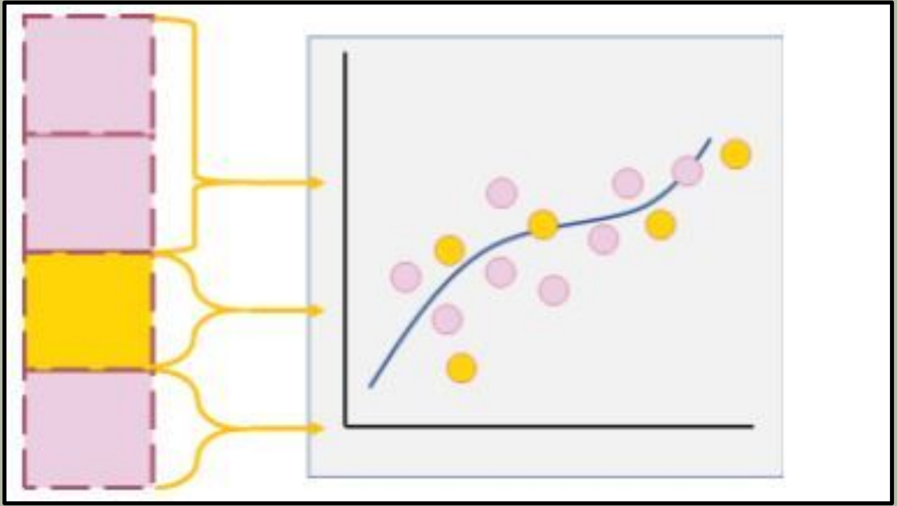
Cross validation - Idea



Training and testing with our portioned dataset

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation - Idea



We then choose a different portion to test on and use the other portions for training. Then, the model performance is re-evaluated with the results obtained from the new portioned dataset to get better results.

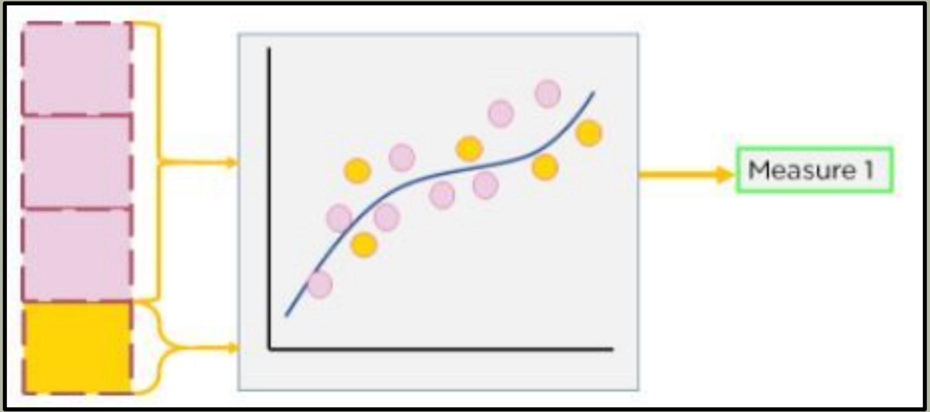
Training and testing on new portions

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation - Steps

Step 1: Split the data into train and test sets and evaluate the model's performance

The first step involves partitioning our dataset and evaluating the partitions. The output measure of accuracy obtained on the first partitioning is noted.



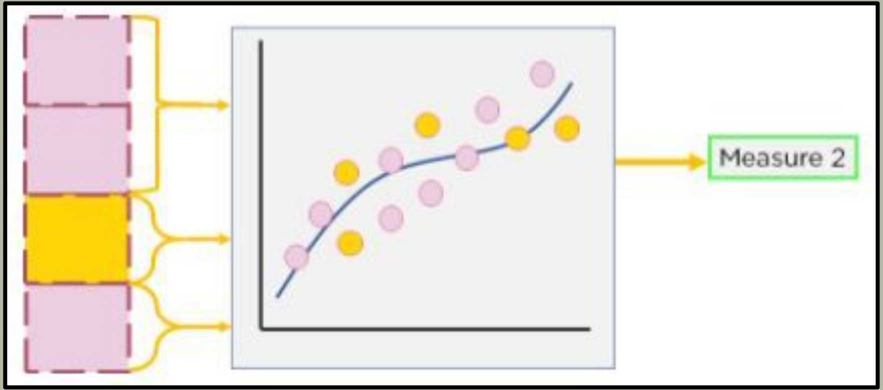
cross-validation partitioning of the dataset

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation - Steps

Step 2: Split the data into new train and test sets and re-evaluate the model's performance

After evaluating one portion of the dataset, we choose a different portion to test and train on. The output measure obtained from this new training and testing dataset is again noted.

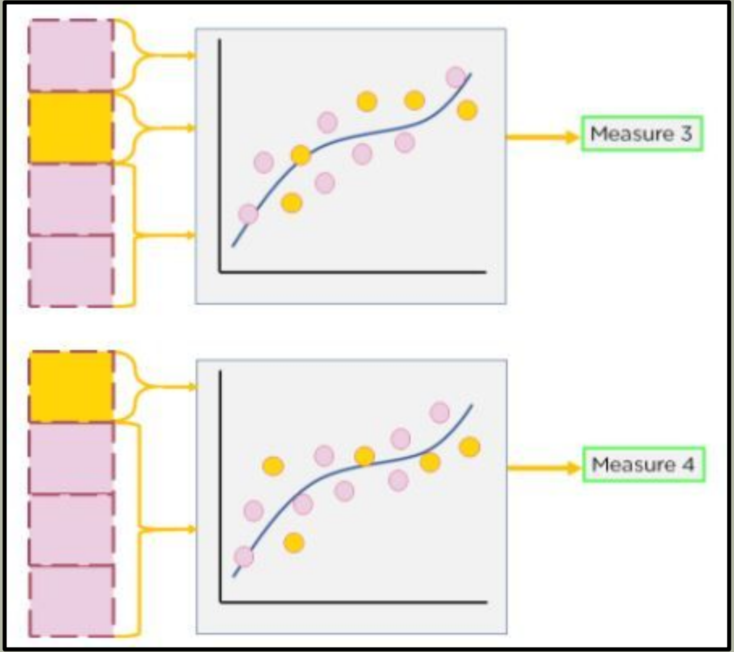


cross-validation revaluation on new portions

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation - Steps

Step 2: This step is repeated multiple times until the model has been trained and evaluated on the entire dataset.

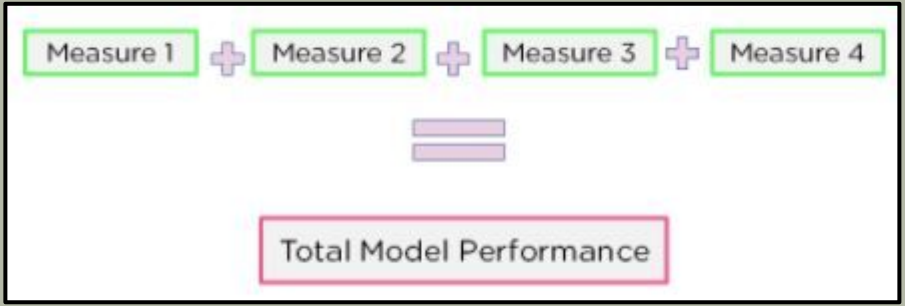


Repeating Step 2 of cross-validation

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation - Steps

Step 3: To get the actual performance metric the average of all measures is taken



Getting model performance

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation - Models (Leave-One-Out Cross Validation)

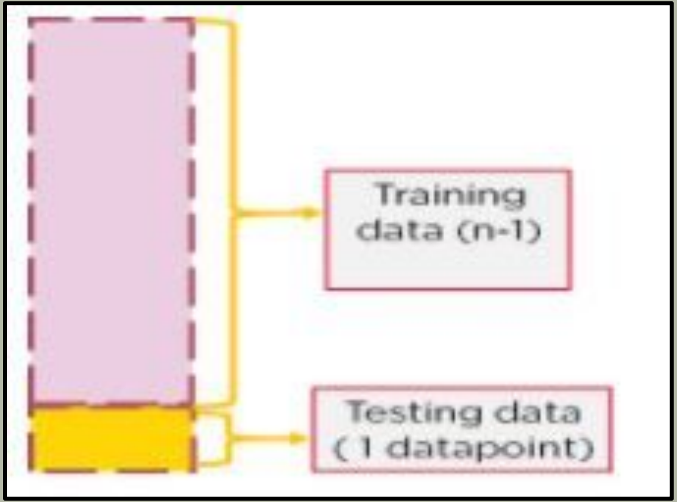
In LOOCV, instead of leaving out a portion of the dataset as testing data, we select one data point as the test data. The rest of the dataset will be used for training and the single data point will be used to predict after training.

Consider a dataset with N points. N-1 will be the training set and 1 point will be the testing set.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Cross validation - Models (Leave-One-Out Cross Validation)

Consider a dataset with N points. N-1 will be the training set and 1 point will be the testing set.

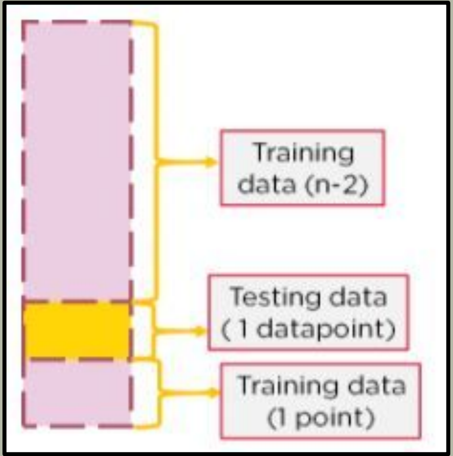


Splitting a dataset for LOOCV

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

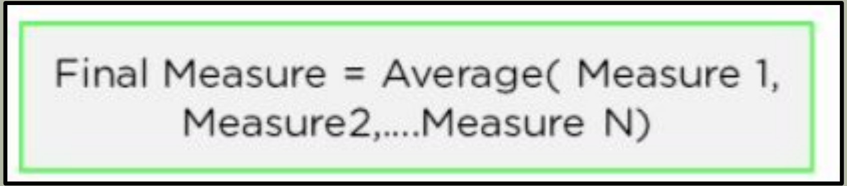
Cross validation - Models (Leave-One-Out Cross Validation)

Another point will be chosen as the testing data and the rest of the points will be training. This will repeat for the rest of the dataset, i.e.: N times.



Selecting another point a testing data

The final performance measure will be the average of the measures for all n iterations.



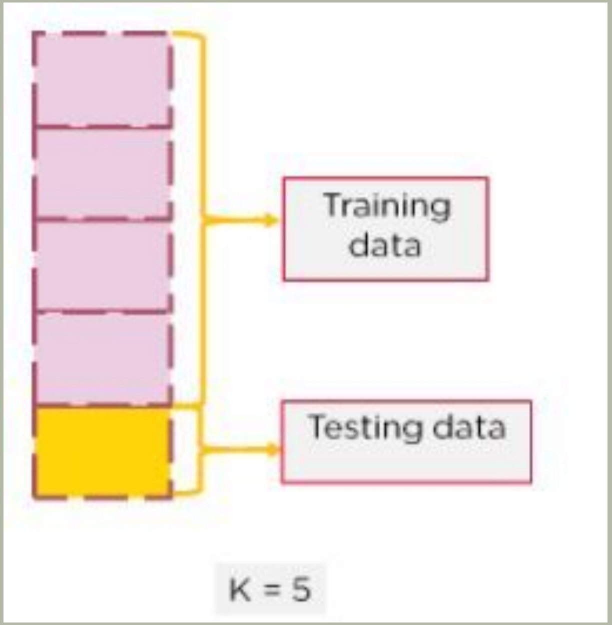
Final Measure = Average(Measure 1, Measure2,....Measure N)

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

K - Fold Cross Validation

In K-fold cross-validation, K refers to the number of portions the dataset is divided into. K is selected based on the size of the dataset.

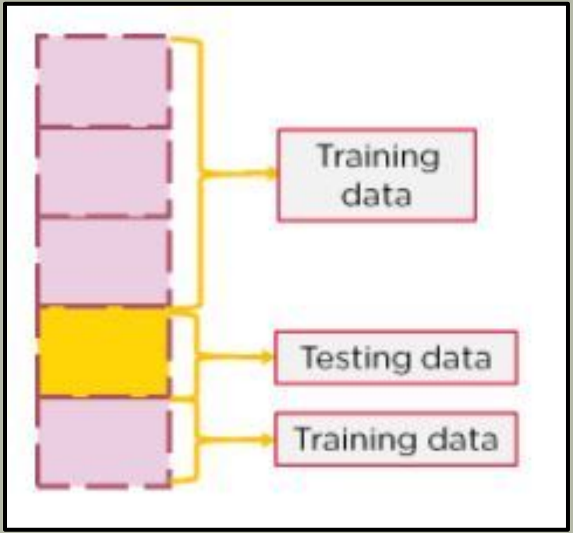
The dataset is split into k portions one section is for testing and the rest for training.



PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

K - Fold Cross Validation

Another section will be chosen for testing and the remaining section will be for training. This will continue K number of times until all sections have been used as a testing set once.

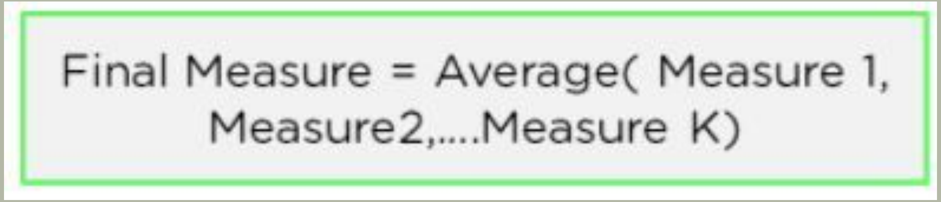


Selecting a different dataset portion for K-Fold CV

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

K - Fold Cross Validation

The final performance measure will be the average of the output measures of the K iterations



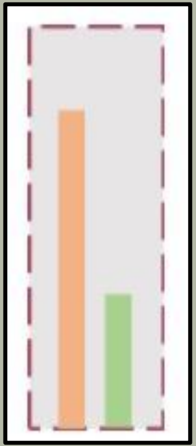
Final Measure = Average(Measure 1, Measure2,....Measure K)

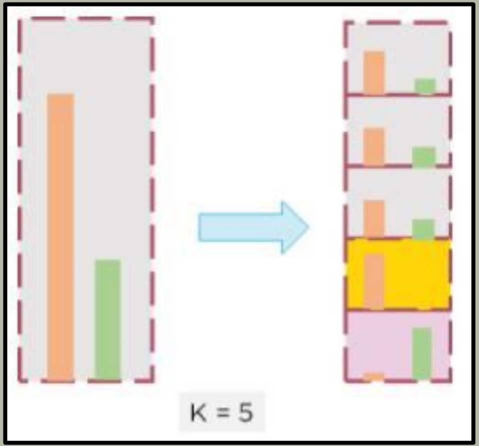
Final accuracy using K-fold

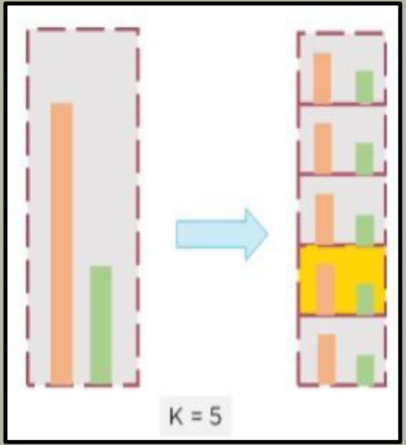
PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Stratified K - Fold Cross Validation

while partitioning the data, some testing sets will include instances of minority classes while others will not. When this happens, our accuracy will not properly reflect how well minority classes are being predicted. To overcome this, The data is split so that each portion has the same percentage of all the different classes that exist in the dataset.







PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Bias in ML

Bias is basically how far we have predicted the value from the actual value. We say the bias is too high if the average predictions are far off from the actual values.

A high bias will cause the algorithm to miss a dominant pattern or relationship between the input and output variables.

When the bias is too high, it is assumed that the model is quite simple and does not fathom the complexity of the data set to determine the relationship and thus, causing underfitting.

Variance in ML

On an independent, unseen data set or a validation set. When a model does not perform as well as it does with the trained data set, there is a possibility that the model has a variance. It basically tells how scattered the predicted values are from the actual values.

A high variance in a data set means that the model has trained with a lot of noise and irrelevant data. Thus causing overfitting in the model. When a model has high variance, it becomes very flexible and makes wrong predictions for new data points. Because it has tuned itself to the data points of the training set.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Variance in ML

Let the variable that we are predicting to be Y and the other independent variables to be X. Now let us assume there is a relationship between the two variables such that:

Y = f(X) + e

In the above equation, Here e is the estimated error with a mean value 0. When we make a classifier using algorithms like linear regression, SVM, etc, the expected squared error at point x will be:

err(x) = Bias2 + Variance + irreducible error

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

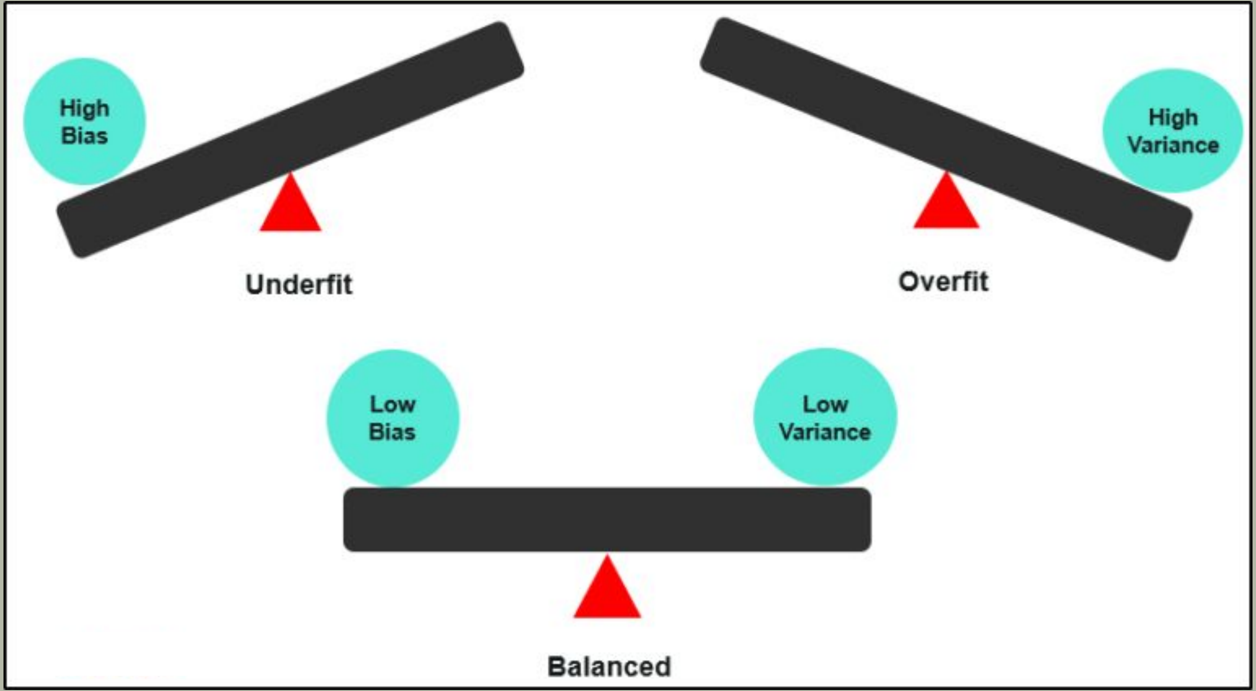
Variance impact on ML model

Let the variable that we are predicting to be Y and the other independent

1. High Variance-High Bias – The model is inconsistent and also inaccurate on average
2. Low Variance-High Bias – Models are consistent but low on average
3. High Variance-Low Bias – Somewhat accurate but inconsistent on averages
4. Low Variance-Low Bias – It is the ideal scenario, the model is consistent and accurate on average.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Variance impact on ML model



PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Bias Variance Trade-off

Finding the right balance between the bias and variance of the model is called the Bias-Variance trade-off. It is basically a way to make sure the model is neither overfitted or underfitted in any case.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Bias Variance Trade-off

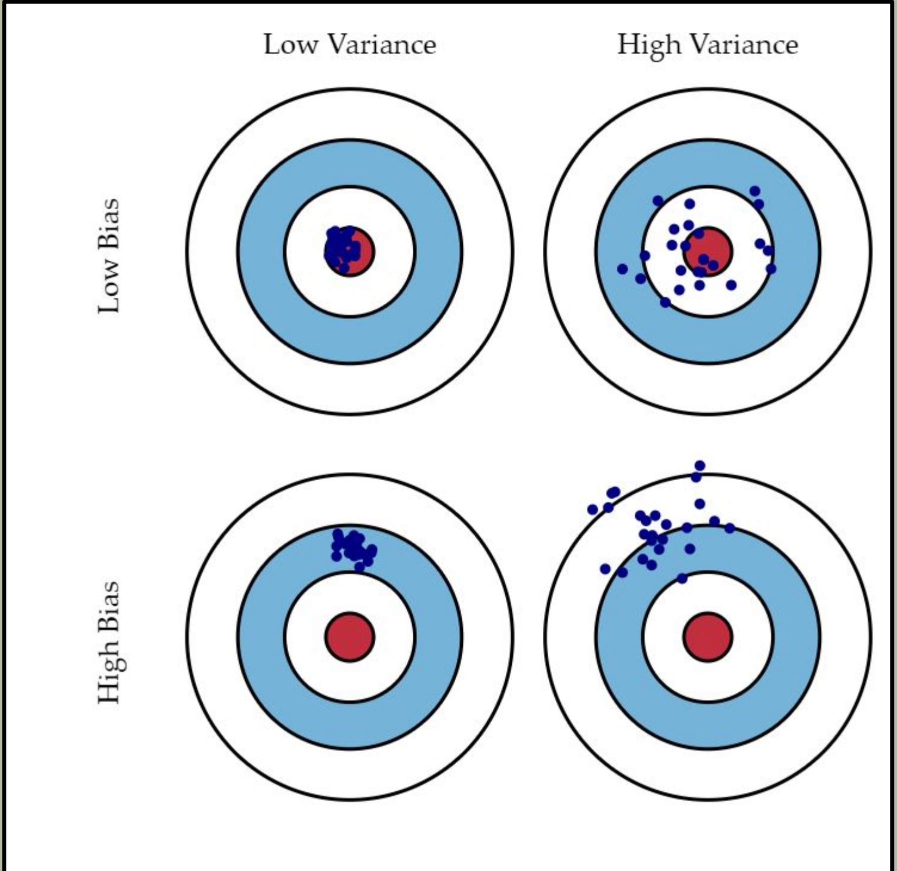


Fig. 1 Graphical illustration of bias and variance.

If the model is too simple and has very few parameters, it will suffer from high bias and low variance.
On the other hand, if the model has a large number of parameters, it will have high variance and low bias.

This trade-off should result in a perfectly balanced relationship between the two. Ideally, low bias and low variance is the target for any Machine Learning model.

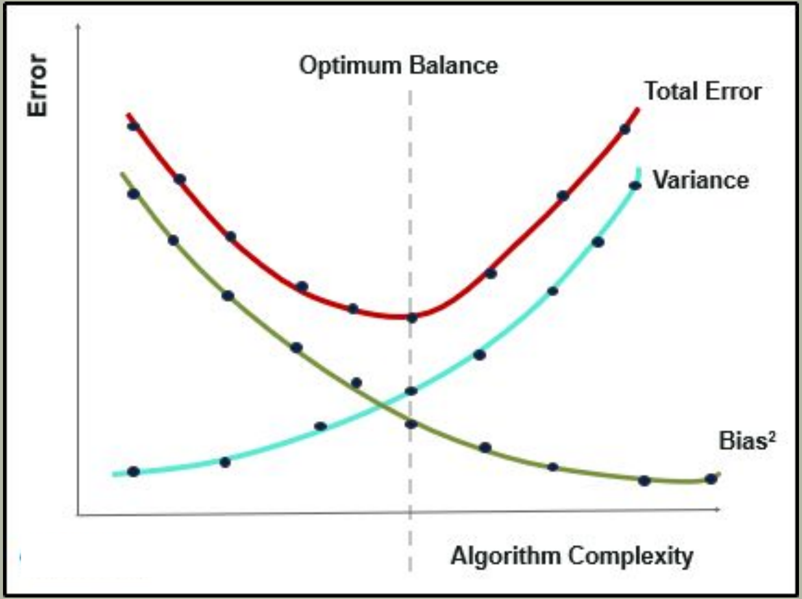
PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Bias Variance Trade-off - Total Error

In any Machine Learning model, a good balance between the bias and variance serves as a perfect scenario in terms of predictive accuracy and avoiding overfitting, underfitting altogether.

An optimal balance between the bias and variance, in terms of algorithm complexity, will ensure that the model is never overfitted or under fitted at all.

The mean squared error in a statistical model is considered as the sum of squared bias and variance and variance of error. All this can be put inside a total error where we have bias, variance and irreducible error in a model.



PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

The Bootstrap

Bootstrap Sampling: It is a method in which we take a sample data repeatedly with replacement from a data set to estimate a population parameter. It is used to determine various parameters of a population.

A bootstrap plot is a graphical representation of the distribution of a statistic calculated from a sample of data. It is often used to visualize the variability and uncertainty of a statistic, such as the mean or standard deviation, by showing the distribution of the statistic over many bootstrapped samples of the data.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

The Bootstrap

In a bootstrap plot, the x-axis represents the values of the statistic and the y-axis represents the frequency of those values. A line is plotted for each bootstrapped sample, with the height of the line indicating the frequency of the statistic value in that sample. The distribution of the lines represents the distribution of the statistic over the bootstrapped samples.

The bootstrap plot is a powerful tool for understanding the uncertainty in a statistic, especially when the underlying distribution of the data is unknown or complex. It can also be used to generate confidence intervals for a statistic and to compare the distributions of different statistics.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

The Bootstrap

Bootstrap plot: It is a graphical method used to measure the uncertainty of any desired statistical characteristic of a population. It is an alternative to the confidence interval. (also a mathematical method used for calculation of a statistic).

Structure

1. x-axis: Subsample number.
2. y-axis: Computed value of the desired statistic for a given subsample.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

The Bootstrap

Need for a Bootstrap plot:

Suppose, we have 5000 people in a park, and we need to find the average weight of the whole population. It is not feasible to measure the weight of each individual and then take an average of that. This is where bootstrap sampling comes into the picture.

What we do is, we take groups of 5 people randomly from the population and find its mean. We do the same process say 8-10 times. This way, we get a good estimate of the average weight of the population more efficiently.

Commonly, we can calculate the uncertainty of a statistic of a population mathematically, using confidence intervals. However, in many cases, the uncertainty formula that is derived is mathematically intractable. In such cases, we use the Bootstrap plot.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

The Bootstrap

Intuition:

Say we have a sample data of 3000 randomly generated uniform numbers. We take out a sub-sample of 30 numbers and find its mean. We do this again for another random sub-sample and so on.

We plot a bootstrap plot of the above-acquired information and just by looking at it, we can easily give a good estimate about the mean of all the 3000 numbers. There is various other useful information one can get out of a bootstrap plot such as:

- which sub-sample had the lowest variance, or
- which sub-sample creates the narrowest confidence interval, etc.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

The Bootstrap

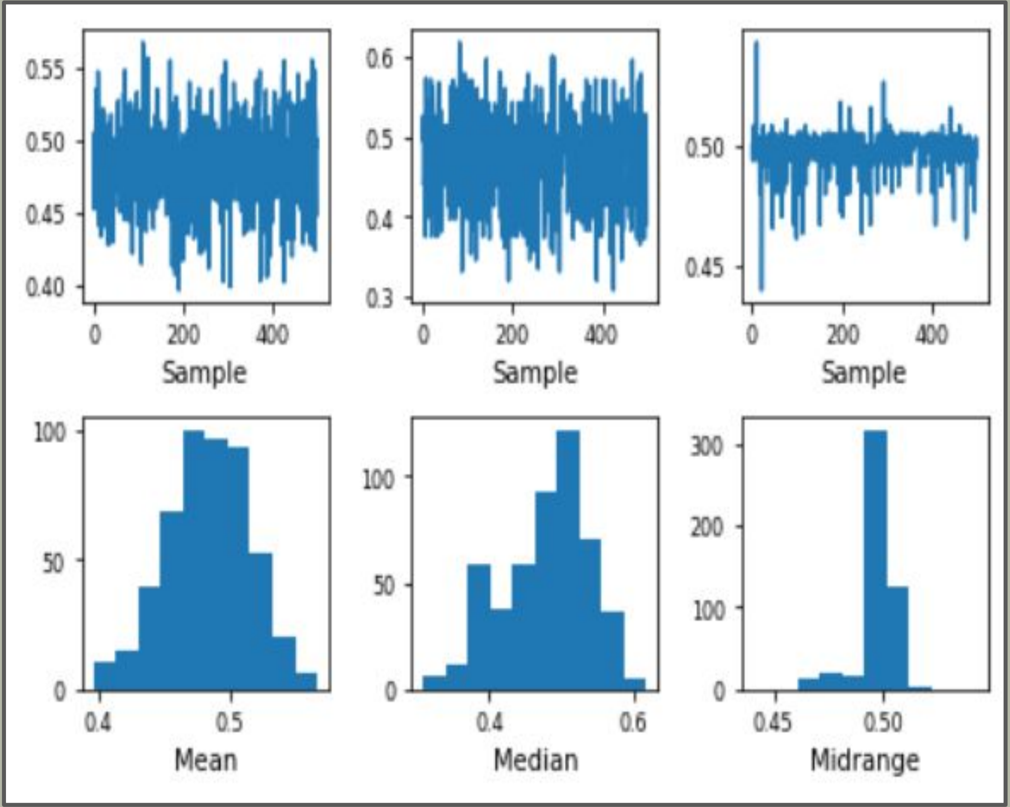
Intuition:

```
import pandas as pd

import numpy as np

s = pd.Series(np.random.uniform(size=500))

pd.plotting.bootstrap_plot(s)
```



PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

The Bootstrap

Limitation:

- The bootstrap plot gives an estimation of the required information from the population, not the exact values.
- It is highly dependent on the dataset given. It fails to give good results when a lot of subsets have repeated samples.
- The bootstrap plot becomes ineffective when we are obtaining information that is highly dependent on the tail values.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

The Bootstrap

Advantages:

- It is a non-parametric method, which means it does not require any assumptions about the underlying distribution of the data.
- It can be used to estimate standard errors and confidence intervals for a wide range of statistics.
- It can be used to estimate the uncertainty of a statistic even when the sample size is small.
- It can be used to perform hypothesis tests and compare the distributions of different statistics.
- It is widely used in many fields such as statistics, finance, and machine learning.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

The Bootstrap

Disadvantages:

- It can be computationally intensive, especially when working with large datasets.
- It may not be appropriate for all types of data, such as highly skewed or heavy-tailed distributions.
- It may not be appropriate for estimating the uncertainty of statistics that have very large variances.
- It may not be appropriate for estimating the uncertainty of statistics that are not smooth or have very different variances.
- It may not always be a good substitute for other statistical methods like asymptotic methods, when large sample sizes are available.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

ROC curve

ROC - Receiver Operating Characteristic curve is a graph that shows a classification model performance at all classification thresholds.

It is a probability curve that plots two parameters, the True Positive Rate (TPR) against the False Positive Rate (FPR), at different threshold values and separates a so-called 'signal' from the 'noise.'

The ROC curve plots the True Positive Rate against the False Positive Rate at different classification thresholds. If the user lowers the classification threshold, more items get classified as positive, which increases both the False Positives and True Positives.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

ROC curve

The FPR is the ratio of negative instances that are incorrectly classified as positive.

It is equal to one minus the true negative rate, which is the ratio of negative instances that are correctly classified as negative.

The TNR is also called specificity. Hence the ROC curve plots sensitivity (recall) versus 1 – specificity.

To plot the ROC curve, you first need to compute the TPR and FPR for various thres- hold values, using the roc_curve() function:

```
from sklearn.metrics import roc_curve fpr, tpr,
thresholds = roc_curve(y_train_5, y_scores)
```

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

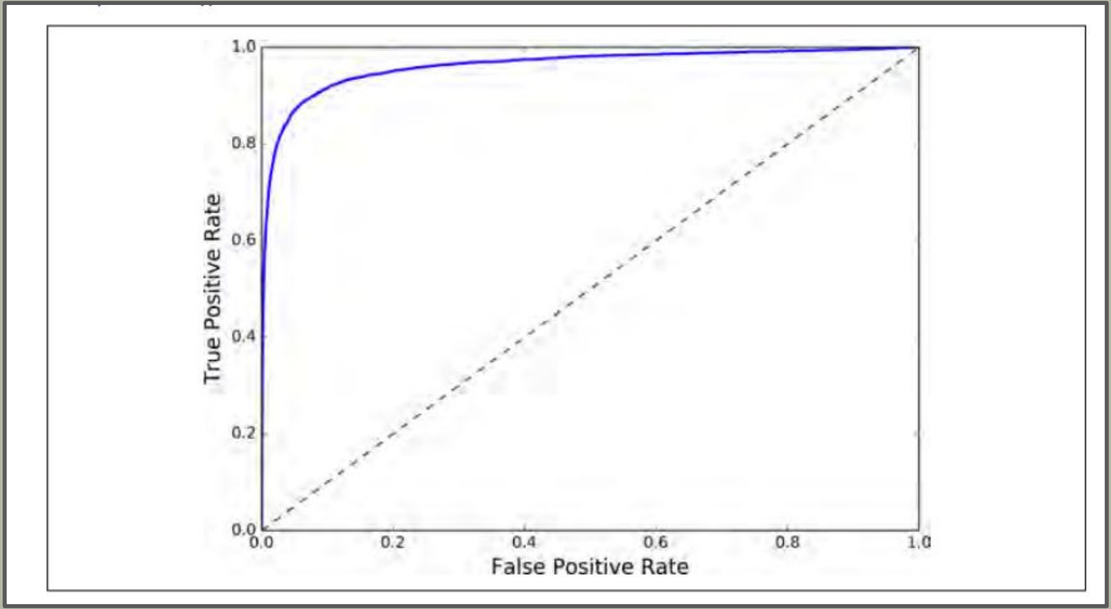
ROC curve

```
from sklearn.metrics import roc_curve fpr, tpr,
thresholds = roc_curve(y_train_5, y_scores)
def plot_roc_curve(fpr, tpr, label=None):
plt.plot(fpr, tpr, linewidth=2, label=label)
plt.plot([0, 1], [0, 1], 'k--')
plt.axis([0, 1, 0, 1])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plot_roc_curve(fpr, tpr)
plt.show()
```

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

ROC curve

Once again there is a tradeoff: the higher the recall (TPR), the more false positives (FPR) the classifier produces. The dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).



ROC curve : AUC (Area under the ROC curve)

AUC measures the whole two-dimensional area located underneath the entire ROC curve from (0,0) to (1,1).

The AUC measures the classifier's ability to distinguish between classes.

It is used as a summary of the ROC curve. The higher the AUC, the better the model can differentiate between positive and negative classes.

AUC supplies an aggregate measure of the model's performance across all possible classification thresholds.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

ROC curve : AUC (Area under the ROC curve)

AUC is a valuable tool for speculating model performance. An excellent model has its AUC close to 1, indicating a good separability measure.

a poor model's AUC leans closer to 0, showing the worst separability measure.

The proximity to 0 means it reciprocates the result, predicting the negative class as positive and vice versa, showing 0's as 1's and 1's as 0's. Finally, if the AUC is 0.5, it shows that the model has no class separation capacity at all.

So, when we have a $0.5<AUC<1$ result, there's a high likelihood that the classifier can distinguish between the positive class values and the negative class values. That's because the classifier can detect more numbers of True Positives and Negatives instead of False Negatives and Positives.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

ROC curve

Sensitivity: Sensitivity, also termed "recall," is the metric that shows a model's ability to predict the true positives of all available categories.

It shows what proportion of the positive class was classified correctly. For example, when trying to figure out how many people have the flu, sensitivity, or True Positive Rate, measures the proportion of people who have the flu and were correctly predicted as having it.

Here's how to mathematically calculate sensitivity:

Sensitivity = (True Positive)/(True Positive + False Negative)

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

ROC curve

Specificity: The specificity metric Specificity evaluates a model's ability to predict true negatives of all available categories.

It shows what proportion of the negative class was classified correctly.

For example, specificity measures the proportion of people who don't have the flu and were correctly predicted as not suffering from it in our flu scenario.

Here’s how to calculate specificity:

Specificity = (True Negative)/(True Negative + False Positive)

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

ROC curve for multiclass classification

We can use the One vs. ALL methodology to plot the N number of AUC ROC Curves for N number classes when using a multi-class model.

for example, you have three classes named 0, 1, and 2. You will have one ROC for 0 that’s classified against 1 and 2, another ROC for 1, which is classified against 0 and 2, and finally, the third one of 2 classified against 0 and 1.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Confusion matrix

Background :

Classification Models have multiple categorical outputs. Most error measures will calculate the total error in our model, but we cannot find individual instances of errors in our model. The model might misclassify some categories more than others, but we cannot see this using a standard accuracy measure.

Furthermore, suppose there is a significant class imbalance in the given data.

In that case, i.e., a class has more instances of data than the other classes, a model might predict the majority class for all cases and have a high accuracy score; when it is not predicting the minority classes. This is where confusion matrices are useful.

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Confusion matrix

Concept :

A confusion matrix presents a table layout of the different outcomes of the prediction and results of a classification problem and helps visualize its outcomes.

It plots a table of all the predicted and actual values of a classifier.

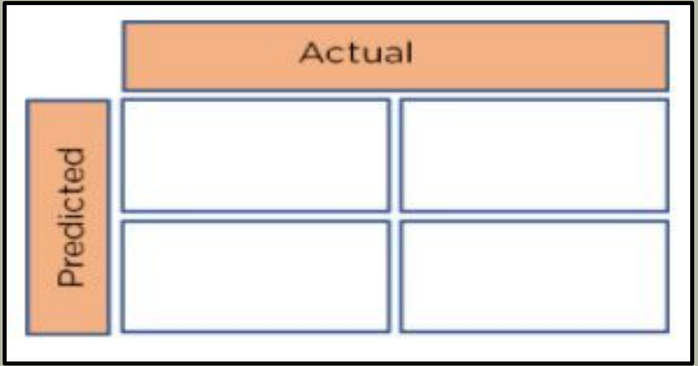


Figure 1: Basic layout of a Confusion Matrix

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Confusion matrix

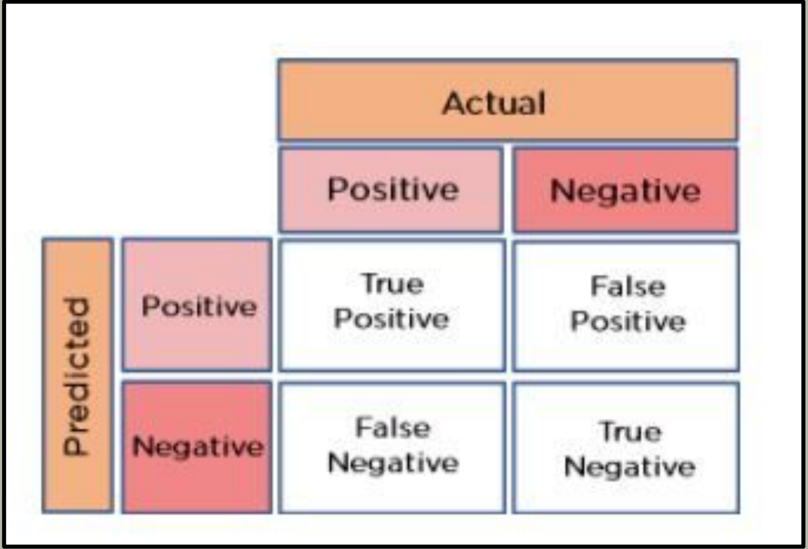
2 * 2 confusion matrix :

True Positive: The number of times our actual positive values are equal to the predicted positive. You predicted a positive value, and it is correct.

False Positive: The number of times our model wrongly predicts negative values as positives. You predicted a negative value, and it is actually positive.

True Negative: The number of times our actual negative values are equal to predicted negative values. You predicted a negative value, and it is actually negative.

False Negative: The number of times our model wrongly predicts negative values as positives. You predicted a negative value, and it is actually positive.



		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Confusion Matrix

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Confusion matrix Metrics

2 * 2 confusion matrix :

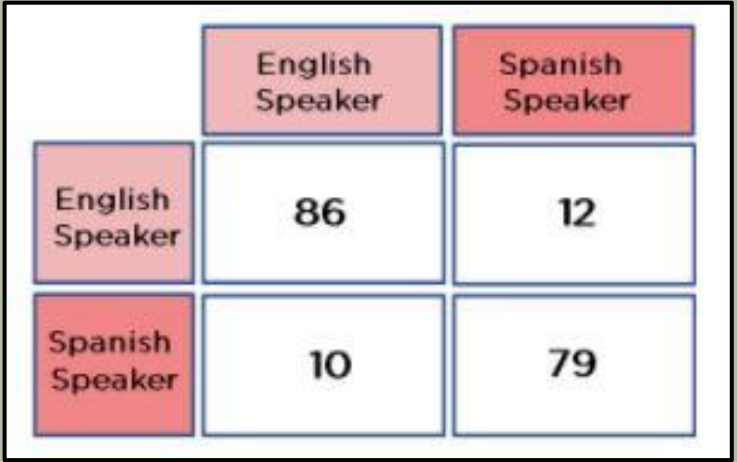
Consider a confusion matrix made for a classifier that classifies people based on whether they speak English or Spanish.

True Positives (TP) = 86

True Negatives (TN) = 79

False Positives (FP) = 12

False Negatives (FN) = 10



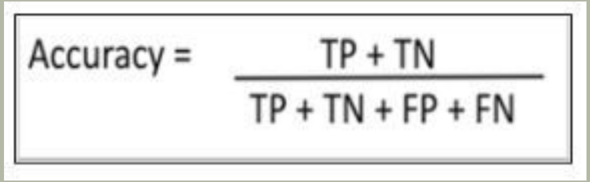
	English Speaker	Spanish Speaker
English Speaker	86	12
Spanish Speaker	10	79

Confusion Matrix for a classifier

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Confusion matrix Metrics

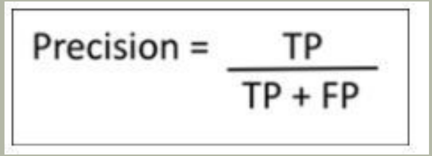
Accuracy: The accuracy is used to find the portion of correctly classified values. It tells us how often our classifier is right. It is the sum of all true values divided by total values.



$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

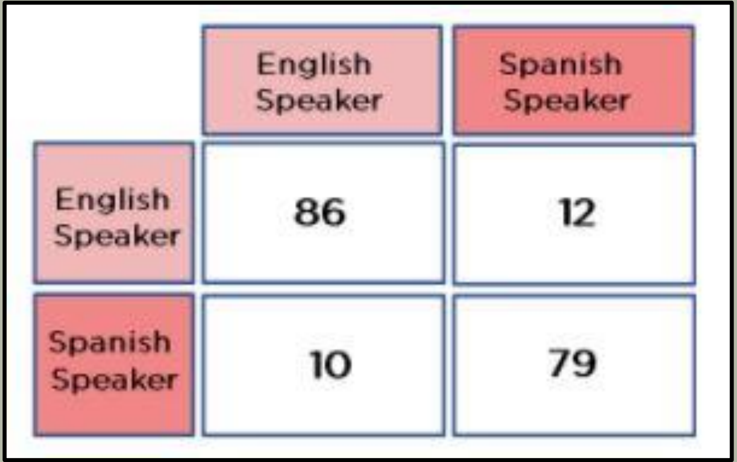
Accuracy = (86 +79) / (86 + 79 + 12 + 10) = 0.8823 = 88.23%

Precision: Precision is used to calculate the model's ability to classify positive values correctly. It is the true positives divided by the total number of predicted positive values.



$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision = 86 / (86 + 12) = 0.8775 = 87.75%



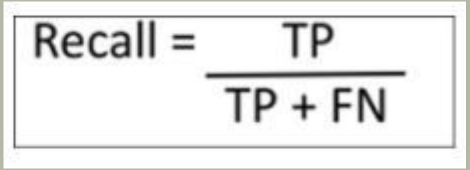
	English Speaker	Spanish Speaker
English Speaker	86	12
Spanish Speaker	10	79

Confusion Matrix for a classifier

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Confusion matrix Metrics

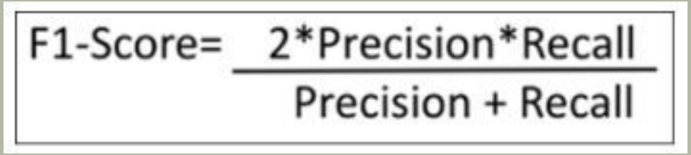
Recall: It is used to calculate the model's ability to predict positive values. "How often does the model predict the correct positive values?". It is the true positives divided by the total number of actual positive values.



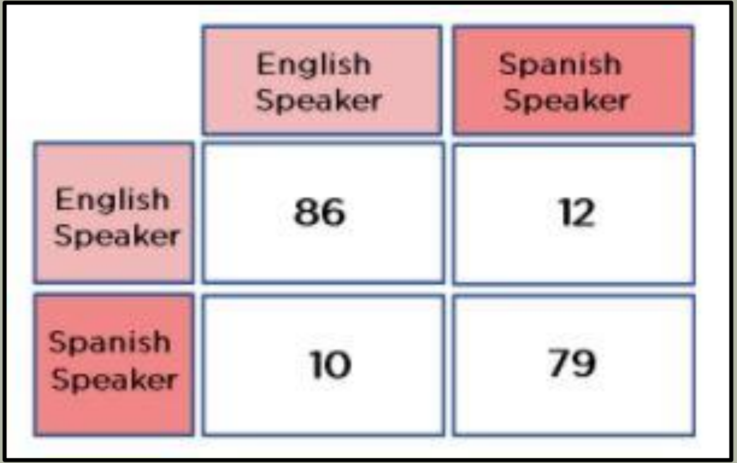
$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall = 86 / (86 + 10) = 0.8983 = 89.83%

F1-Score: It is the harmonic mean of Recall and Precision. It is useful when you need to take both Precision and Recall into account.



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



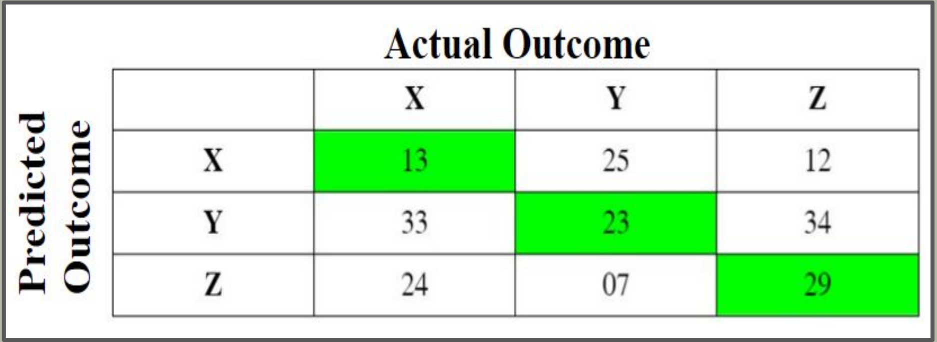
	English Speaker	Spanish Speaker
English Speaker	86	12
Spanish Speaker	10	79

Confusion Matrix for a classifier

F1-Score = (2* 0.8775 * 0.8983) / (0.8775 + 0.8983) = 0.8877 = 88.77%

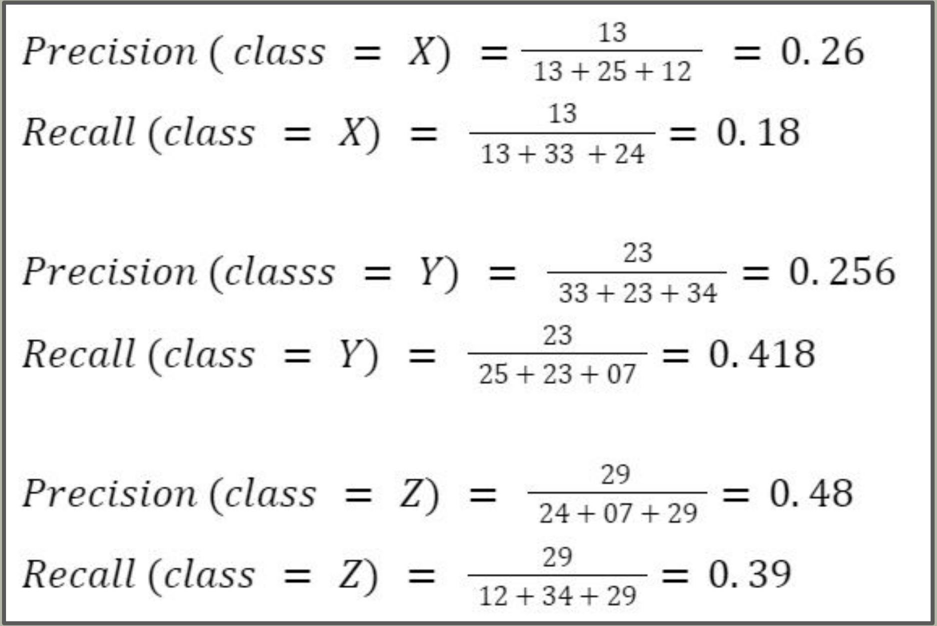
PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Confusion matrix Metrics



Predicted Outcome	Actual Outcome		
	X	Y	Z
X	🟩 13	25	12
Y	33	🟩 23	34
Z	24	07	🟩 29

Confusion Matrix with 200 samples



$$Precision\,(\,class \;=\; X) \;=\; \frac{13}{13+25+12} \;=0.26$$

$$Recall\,(class \;=\; X) \;=\; \frac{13}{13+33\;+24} = 0.18$$

$$Precision\,(classs \;=\; Y) \;=\; \frac{23}{33+23+34} = 0.256$$

$$Recall\,(class \;=\; Y) \;=\; \frac{23}{25+23+07} = 0.418$$

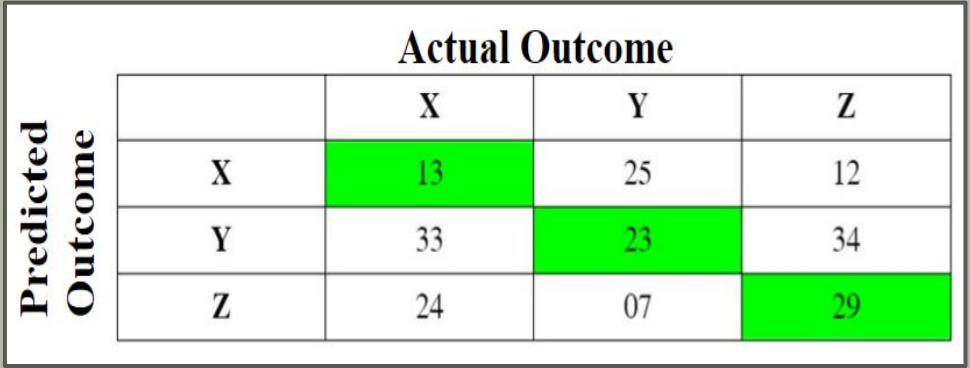
$$Precision\,(class \;=\; Z) \;=\; \frac{29}{24+07+29} = 0.48$$

$$Recall\,(class \;=\; Z) \;=\; \frac{29}{12+34+29} = 0.39$$

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM

Confusion matrix Metrics

f1 score is the evaluation metric that is used to evaluate the performance of the machine learning model.
It uses both precision and Recall, that makes it best for unbalanced dataset.



		Actual Outcome		
		X	Y	Z
Predicted Outcome	X	13	25	12
	Y	33	23	34
	Z	24	07	29

Confusion Matrix with 200 samples

$$F1\,score\,(class\ =\ X) = \frac{2*(0.26)*(0.18)}{0.26+0.18} = 0.213$$

$$F1\,score\,(class\ =\ Y) = \frac{2*(0.256)*(0.418)}{0.256+0.418} = 0.317$$

$$F1\,score\,(class\ =\ Z) = \frac{2*(0.48)*(0.39)}{0.48+0.39} = 0.430$$

PROF.TRUPESH PATEL
TRUPESH1991@GMAIL.COM