Data Pre-Processing

Missing Values

Missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analyzed the behavior and relationship with other variables correctly. It can lead to wrong prediction or classification.

Name	Weight	Gender	Play Cricket/ Not
Mr. Amit	58	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55		Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57		Y
Mr. Kunal	57	M	N

Gender	#Students	#Play Cricket	%Play Cricket
F	2	1	50%
M	4	2	50%
Missing	2	2	100%

Name	Weight	Gender	Play Cricket/ Not
Mr. Amit	58	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55	F	Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57	F	Y
Mr. Kunal	57	M	N

Gender	#Students	#Play Cricket	%Play Cricket
F	4	3	75%
M	4	2	50%

Name	Weight	Gender	Play Cricket/ Not
Mr. Amit	58	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55		Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57		Y
Mr. Kunal	57	M	N

Gender	#Students	#Play Cricket	%Play Cricket
F	2	1	50%
M	4	2	50%
Missing	2	2	100%

Name	Weight	Gender	Play Cricket/ Not
Mr. Amit	58	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55	F	Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57	F	Y
Mr. Kunal	57	M	N

Gender	#Students	#Play Cricket	%Play Cricket
F	4	3	75%
M	4	2	50%

Notice the missing values in the image shown above: In the left scenario, we have not treated missing values. The inference from this data set is that the chances of playing cricket by males is higher than females. On the other hand, if you look at the second table, which shows data after treatment of missing values (based on gender), we can see that females have higher chances of playing cricket compared to males.

Why Data could have missing values?

We looked at the importance of treatment of missing values in a dataset. Now, let's identify the reasons for occurrence of these missing values. They may occur at two stages:

1. **Data Extraction**: It is possible that there are problems with extraction process. In such cases, we should double-check for correct data with data guardians. Some hashing procedures can also be used to make sure data extraction is correct. Errors at data extraction stage are typically easy to find and can be corrected easily as well.
2. **Data collection**: These errors occur at time of data collection and are harder to correct. They can be categorized in four types:

2.1 Missing completely at random: This is a case when the probability of missing variable is same for all observations. For example: respondents of data collection process decide that they will declare their earning after tossing a fair coin. If an head occurs, respondent declares his / her earnings & vice versa. Here each observation has equal chance of missing value.

2.2 Missing at random: This is a case when variable is missing at random and missing ratio varies for different values / level of other input variables. For example: We are collecting data for age and female has higher missing value compare to male.

2.3 Missing that depends on unobserved predictors: This is a case when the missing values are not random and are related to the unobserved input variable. For example: In a medical study, if a particular diagnostic causes discomfort, then there is higher chance of drop out from the study. This missing value is not at random unless we have included "discomfort" as an input variable for all patients.

2.4 Missing that depends on the missing value itself: This is a case when the probability of missing value is directly correlated with missing value itself. For example: People with higher or lower income are likely to provide non-response to their earning.

Methods to treat Missing values

1. **Deletion:** In this technique we simply delete the observations wherever there are missing values, the threshold of performing this technique is only useful when there are very few number of missing observations. By deleting the missing observation the deleted values will have minimal impact on overall data.
2. **Mean/Mode/Median Imputation:** Imputation is a method to fill in the missing values with estimated ones. The objective is to employ known relationships that can be identified in the valid values of the data set to assist in estimating the missing values. Mean / Mode / Median imputation is one of the most frequently used methods. It consists of replacing the missing data for a given attribute by the mean or median (quantitative attribute) or mode (qualitative attribute) of all known values of that variable. In short the missing values in the observations are replaced with the mean values of the particular variable

Prediction Model: Prediction model is one of the sophisticated method for handling missing data. Here, we create a predictive model to estimate values that will substitute the missing data. In this case, we divide our data set into two sets: One set with no missing values for the variable and another one with missing values. First data set become training data set of the model while second data set with missing values is test data set and variable with missing values is treated as target variable. Next, we create a model to predict target variable based on other attributes of the training data set and populate missing values of test data set. We can use regression, ANOVA, Logistic regression and various modeling technique to perform this. There are 2 drawbacks for this approach: The model estimated values are usually more well-behaved than the true values If there are no relationships with attributes in the data set and the attribute with missing values, then the model will not be precise for estimating missing values.

KNN Imputation: In this method of imputation, the missing values of an attribute are imputed using the given number of attributes that are most similar to the attribute whose values are missing. The similarity of two attributes is determined using a distance function. It is also known to have certain advantage & disadvantages.

1. **Advantages**
    - **1.1** k-nearest neighbor can predict both qualitative & quantitative attributes
    - **1.2** Creation of predictive model for each attribute with missing data is not required
    - **1.3** Attributes with multiple missing values can be easily treated
    - **1.4** Correlation structure of the data is taken into consideration
2. **Disadvantage**
    - **2.1** KNN algorithm is very time-consuming in analyzing large database. It searches through all the dataset looking for the most similar instances.
    - **2.2** Choice of k-value is very critical. Higher value of k would include attributes which are significantly different from what we need whereas lower value of k implies missing out of significant attributes.

Iterative Imputer:

Iterative Imputer works in very simple manner, basically a Machine Learning Model executes on the data where the column that have missing values becomes the Dependent Variable and other features becomes Independent Variables, in Iterative Imputer algorithm automatically takes care of the Separation of Dependent and Independent variables. Wherever the missing values comes it gets Imputed with the predicted values. In Iterative Imputer any Machine Learning algorithm can gets executed based on the value that has missing values i.e Categorical/Continuous columns.

Data with Missing Values

	Section	Test1	Test2	Final
0	1.0	94	91	87
1	2.0	51	65	91
2	1.0	95	97	97
3	NaN	63	75	80
4	2.0	80	76	71
5	NaN	92	40	86
6	1.0	75	78	72
7	1.0	94	91	87
8	NaN	51	65	91
9	1.0	95	97	97
10	2.0	63	75	80
11	2.0	80	76	71
12	1.0	92	40	86
13	NaN	75	78	72

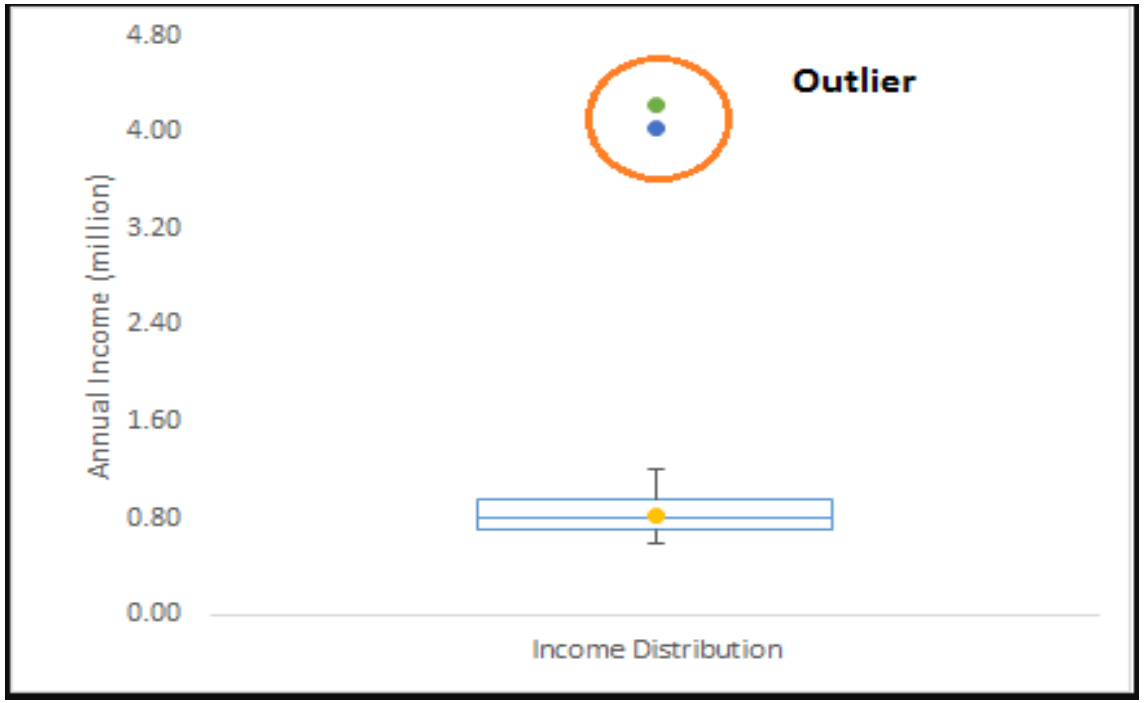
Data with Predicted Missing Values

	Section	Test1	Test2	Final
0	1.0	94.0	91.0	87.0
1	2.0	51.0	65.0	91.0
2	1.0	95.0	97.0	97.0
3	2.0	63.0	75.0	80.0
4	2.0	80.0	76.0	71.0
5	1.0	92.0	40.0	86.0
6	1.0	75.0	78.0	72.0
7	1.0	94.0	91.0	87.0
8	2.0	51.0	65.0	91.0
9	1.0	95.0	97.0	97.0
10	2.0	63.0	75.0	80.0
11	2.0	80.0	76.0	71.0
12	1.0	92.0	40.0	86.0
13	2.0	75.0	78.0	72.0

Outlier – Detection and Treatment

Outlier is a commonly used terminology by analysts and data scientists as it needs close attention else it can result in wildly wrong estimations. Simply speaking, Outlier is an observation that appears far away and diverges from an overall pattern in a sample.

Let's take an example, we do customer profiling and find out that the average annual income of customers is $0.8 million. But, there are two customers having annual income of $4 and $4.2 million. These two customers annual income is much higher than rest of the population. These two observations will be seen as Outliers.



Cause of Outliers

Whenever we come across outliers, the ideal way to tackle them is to find out the reason of having these outliers. The method to deal with them would then depend on the reason of their occurrence. Causes of outliers can be classified in two broad categories:

1. Artificial(Error) / Non-natural
2. Natural.

Various types of outliers in more details:

1. **Data Entry Errors:** Human errors such as errors caused during data collection, recording, or entry can cause outliers in data. For example: Annual income of a customer is $100,000. Accidentally, the data entry operator puts an additional zero in the figure. Now the income becomes $1,000,000 which is 10 times higher. Evidently, this will be the outlier value when compared with rest of the population.
2. **Measurement Error:** It is the most common source of outliers. This is caused when the measurement instrument used turns out to be faulty. For example: There are 10 weighing machines. 9 of them are correct, 1 is faulty. Weight measured by people on the faulty machine will be higher / lower than the rest of people in the group. The weights measured on faulty machine can lead to outliers.
3. **Data Processing Error:** Whenever we perform data mining, we extract data from multiple sources. It is possible that some manipulation or extraction errors may lead to outliers in the dataset.

Impact of Outliers on Dataset

Outliers can drastically change the results of the data analysis and statistical modeling. There are numerous unfavorable impacts of outliers in the data set:

1. It increases the error variance and reduces the power of statistical tests
2. If the outliers are non-randomly distributed, they can decrease normality
3. They can bias or influence estimates that may be of substantive interest

Without Outlier	With Outlier
4, 4, 5, 5, 5, 5, 6, 6, 6, 7, 7	4, 4, 5, 5, 5, 5, 6, 6, 6, 7, 7,300
Mean = 5.45	Mean = 30.00
Median = 5.00	Median = 5.50
Mode = 5.00	Mode = 5.00
Standard Deviation = 1.04	Standard Deviation = 85.03

As you can see, data set with outliers has significantly different mean and standard deviation. In the first scenario, we will say that average is 5.45. But with the outlier, average soars to 30. This would change the estimate completely.

How to Detect Outliers

Most commonly used method to detect outliers is visualization. We use various visualization methods, like **Box-plot**, **Histogram**, **Scatter Plot**

How to remove Outliers

Most of the ways to deal with outliers are similar to the methods of missing values like deleting observations, transforming them, binning them, treat them as a separate group, imputing values and other statistical methods. Here, we will discuss the common techniques used to deal with outliers:

1. **Deleting observations:** We delete outlier values if it is due to data entry error, data processing error or outlier observations are very small in numbers. We can also use trimming at both ends to remove outliers.
2. **Replacing with Mean:** We can also treat the outliers with the same as we treated missing values, by doing so we can bring those outliers point within the boundary of dataset.

Feature Engineering

Feature Engineering

Feature engineering is the science (and art) of extracting more information from existing data. You are not adding any new data here, but you are actually making the data you already have more useful.

For example, let's say you are trying to predict foot fall in a shopping mall based on dates. If you try and use the dates directly, you may not be able to extract meaningful insights from the data. This is because the foot fall is less affected by the day of the month than it is by the day of the week. Now this information about day of week is implicit in your data. You need to bring it out to make your model better.

Feature Engineering includes **Missing Value Treatment**, **Outlier's Treatment**, **Dummy Variables** and **Variable Creation**

Variable Creation & Benefits

Feature / Variable creation is a process to generate a new variables / features based on existing variable(s). For example, say, we have date(dd-mm-yy) as an input variable in a data set. We can generate new variables like day, month, year, week, weekday that may have better relationship with target variable. This step is used to highlight the hidden relationship in a variabl

Emp_Code	Gender	Date	New_Day	New_Month	New_Year
A001	Male	21-Sep-11	21	9	2011
A002	Female	27-Feb-13	27	2	2013
A003	Female	14-Nov-12	14	11	2012
A004	Male	07-Apr-13	7	4	2013
A005	Female	21-Jan-11	21	1	2011
A006	Male	26-Apr-13	26	4	2013
A007	Male	15-Mar-12	15	3	2012

Dummy Variables

One of the most common application of dummy variable is to convert categorical variable into numerical variables. Dummy variables are also called Indicator Variables. It is useful to take categorical variable as a predictor in statistical models. Categorical variable can take values 0 and 1. Let's take a variable 'gender'. We can produce two variables, namely, "**Var_Male**" with values 1 (Male) and 0 (No male) and "**Var_Female**" with values 1 (Female) and 0 (No Female). We can also create dummy variables for more than two classes of a categorical variables with n or n-1 dummy variables.

Emp_Code	Gender	Var_Male	Var_Female
A001	Male	1	0
A002	Female	0	1
A003	Female	0	1
A004	Male	1	0
A005	Female	0	1
A006	Male	1	0
A007	Male	1	0

One Hot Encoding

What are Categorical Data?

Categorical data are variables that contain label values rather than numeric values.
The number of possible values is often limited to a fixed set.
Some examples include:
A "pet" variable with the values: "dog" and "cat".
A "color" variable with the values: "red", "green" and "blue".
A "place" variable with the values: "first", "second" and "third".
Each value represents a different category.

Problem with Categorical Data

Some algorithms can work with categorical data directly.
For example, a decision tree can be learned directly from categorical data with no data transform required (this depends on the specific implementation).
Many machine learning algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric. In general, this is mostly a constraint of the efficient implementation of machine learning algorithms rather than hard limitations on the algorithms themselves.
This means that categorical data must be converted to a numerical form. If the categorical variable is an output variable, you may also want to convert predictions by the model back into a categorical form in order to present them or use them in some application.

How to convert Categorical data to Numerical Data:

1. Integer Encoding

As a first step, each unique category value is assigned an integer value.

For example, "*red*" is 1, "*green*" is 2, and "*blue*" is 3.

This is called a label encoding or an integer encoding and is easily reversible.

For some variables, this may be enough.

The integer values have a natural ordered relationship between each other and machine learning algorithms **may** be able to understand and harness this relationship.

For example, ordinal variables like the "**place**" example above would be a good example where a label encoding would be sufficient.

2. **One Hot Encoding**

For categorical variables where no such ordinal relationship exists, the integer encoding is not enough.

In fact, using this encoding and allowing the model to assume a natural ordering between categories may result in poor performance or unexpected results (predictions halfway between categories).

In this case, a one-hot encoding can be applied to the integer representation. This is where the integer encoded variable is removed and a new binary variable is added for each unique integer value.

In the "*color*" variable example, there are 3 categories and therefore 3 binary variables are needed. A "1" value is placed in the binary variable for the color and "0" values for the other colors.

```
1 red,    green,  blue
2 1,      0,      0
3 0,      1,      0
4 0,      0,      1
```

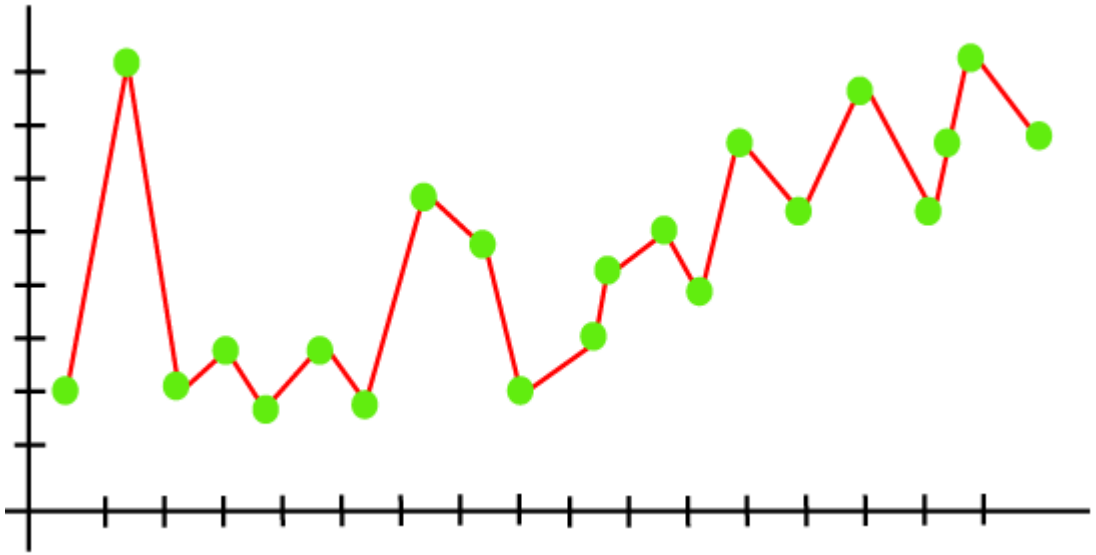
Overfitting – A cause in Machine Learning

Overfitting occurs when our machine learning model tries to cover all the data points or more than the required data points present in the given dataset. Because of this, the model starts caching noise and inaccurate values present in the dataset, and all these factors reduce the efficiency and accuracy of the model. The overfitted model has low bias and high variance.

The chances of occurrence of overfitting increase as much we provide training to our model. It means the more we train our model, the more chances of occurring the overfitted model.

Overfitting is the main problem that occurs in supervised learning.

As we can see from the above graph, the model tries to cover all the data points present in the scatter plot. It may look efficient, but in reality, it is not so. Because the goal of the regression model to find the best fit line, but here we have not got any best fit, so, it will generate the prediction errors.

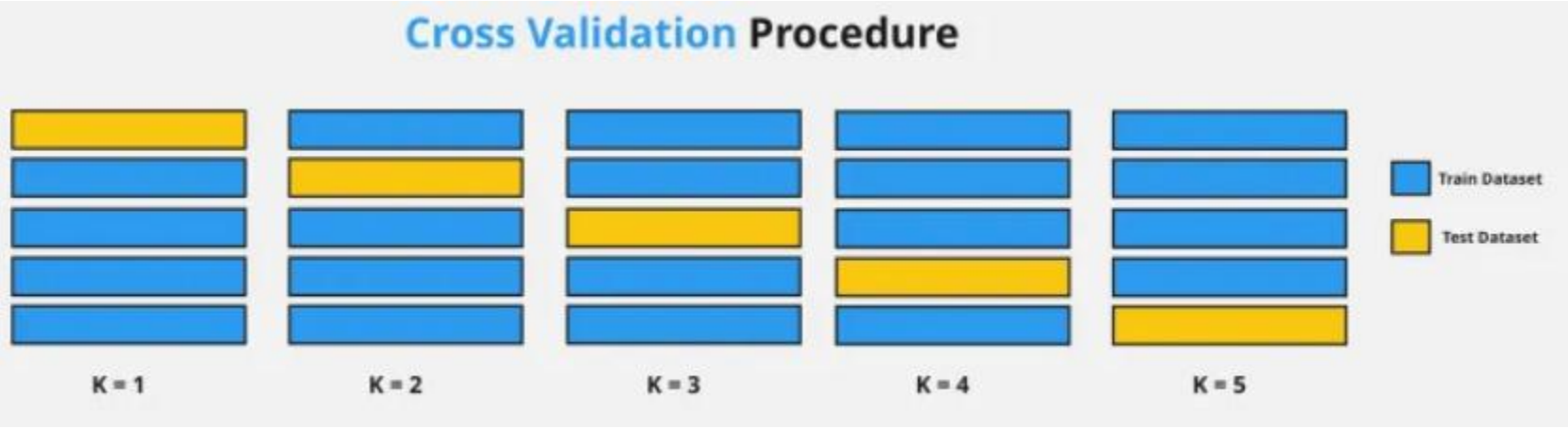


How to avoid Overfitting

1. Cross-Validation
2. Training with more data
3. Removing Features
4. Ensembling

5. Regularization

Cross-Validation: Cross-validation is the best preventive measure against overfitting. It is a smart technique that allows us to utilize our data in a better way.
In the data mining models or machine learning models, separation of data into training and testing sets is an essential part. Minimizing the data discrepancies and better understanding of the machine learning model's properties can be done using similar data for the training and testing subsets.



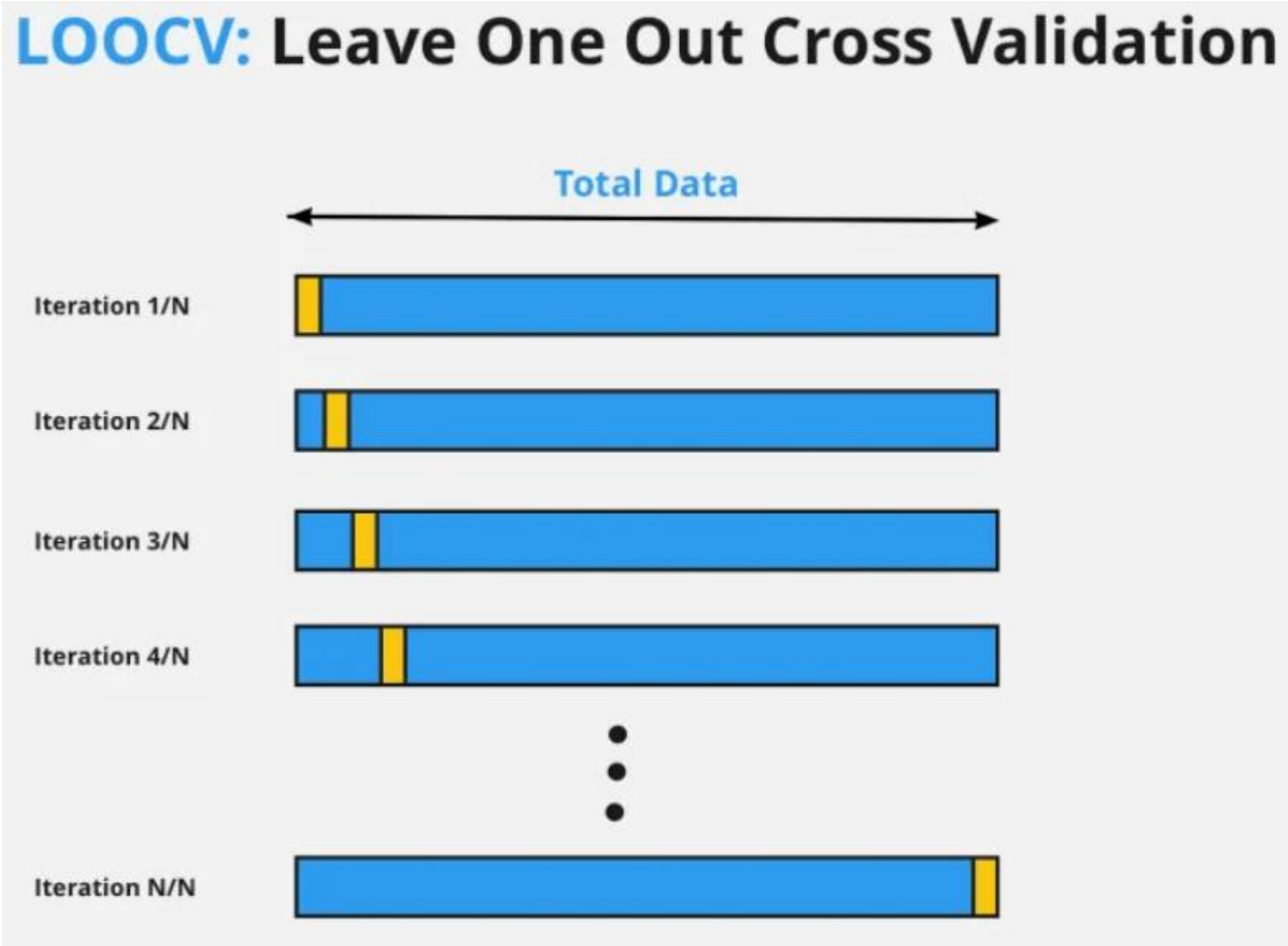
Types of Cross-Validation

Leave **O**ne **O**ut **C**ross **V**alidation (**LOOCV**)

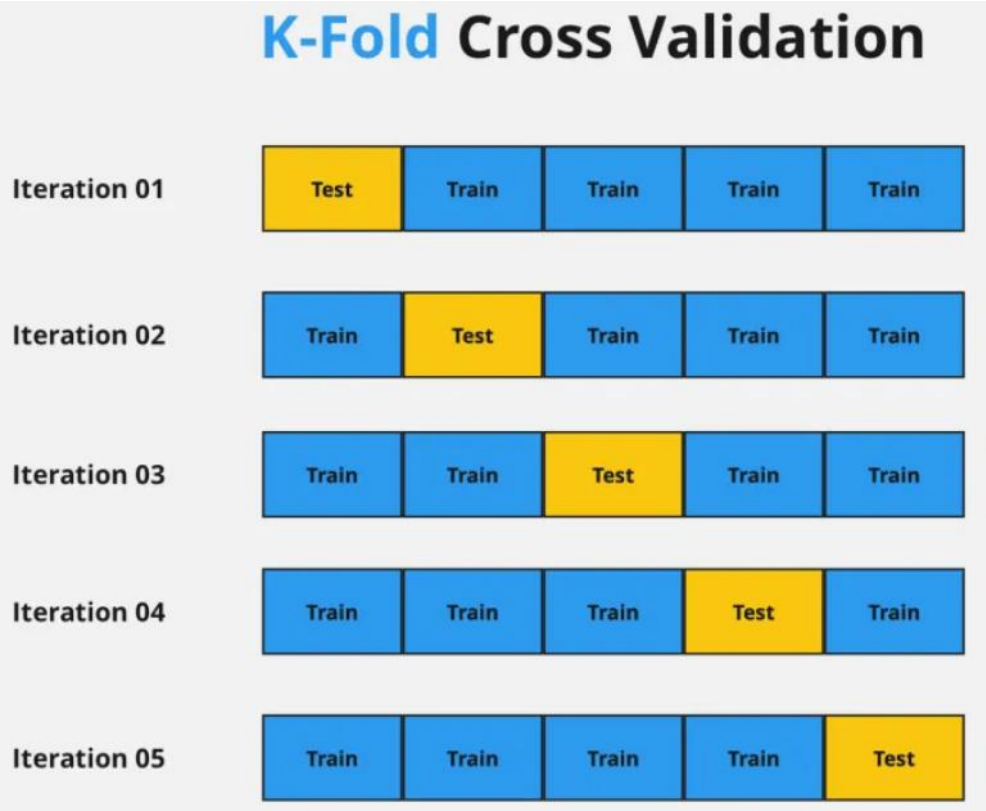
This variation on cross-validation leaves one data point out of the training data. For instance, if there are n data points in the original data sample, then the pieces used to train the model are **n-1**, and **p points** will be used as the validation set.

This cycle is repeated in all of the combinations where the original sample can be separated in such a way. After this, the mean of the error is taken for all trials to give overall effectiveness.

We consider that the number of possible combinations is equal to the number of data points in the original sample represented by **n.**



K-Fold Cross Validation



Suppose we have 1000 records in our dataset. And we select the value of K as 5. So K value means, the number of rounds we perform Training and Testing.
And here, our k value is 5, that means we have to perform 5 round of Training and Testing. Here, our k value is 5. And we have 1000 records. So when we divide 1000 with 5. 1000/5=200
So, 200 will be our Test Data. And the remaining 800 records will be our Training Data. So for the first round, the first 200 records are used for Test Data and the remaining 800 for Training data. This is the first round. So, this model is trained with 800 datasets. And tested on 200 datasets. After that, we got our first Accuracy. Let's say Accuracy 1. In second round, the next 200 records will be our Test data, and the remaining 800 data is our Training data.

Here, our model is trained on 800 records and tested on the next 200 records. And Again we got some accuracy. Let's say Accuracy 2.
In the same way, all rounds happened., and we got accuracies with each round. So after completing all 5 rounds, we got 5 accuracies. The Accuracy 1, Accuracy 2, Accuracy 3, Accuracy 4, and accuracy 5. We can take all Accuracies, and find out the Mean. And that Mean of all 5 accuracies is the actual accuracy of your model.
By doing so, your accuracy will not fluctuate as in Train-Test Split. Moreover, you will get your model minimum accuracy and maximum accuracy.

Advantage and Disadvantage of **K-Fold Cross Validation**

Advantages

- K-fold cross-validation works well on small and large data sets.
- All of our data is used in testing our model, thus giving a fair, well-rounded evaluation metric.
- K-fold cross-validation may lead to more accurate models since we are eventually utilizing our data to build our model.

Disadvantages

- The computing power is high.
- So it may take some time to get feedback on the model's performance in the case of large data sets.
- Slower feedback makes it take longer to find the optimal hyperparameters for the model.

Stratified Cross Validation:

The process of rearranging the data to ensure that each fold is a good representative of the whole is termed stratification.
For instance, in the case of a binary classification problem, each class is comprises of 50% of the data. Let's say the ration is 30% and 70% distribution. So the best practice is to arrange the data so that each class consists of the same 30%
and 70% distribution in every fold.
Note that 30% and 70% ration is
not imbalanced data.