1. What is the definition of a target function? In the sense of a real-life example, express the target function. How is a target function's fitness assessed?

**Introduction to Bootstrap plot**

* Before getting into Bootstrap plot, let us first understand what Bootstrapping (or Bootstrap sampling) is all about.

**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.

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’s 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.

It is important to note that Bootstrap is a resampling technique which is used to estimate the uncertainty of a statistic from a sample, without making any assumptions about the underlying distribution of the data. It can be used to estimate standard errors, confidence intervals, and to perform hypothesis tests.

**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**

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

**Need for a Bootstrap plot:**

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.

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.

**Intuition:**

Let us consider an example and understanding how the Bootstrap plot makes it easier to obtain critical information from a large population. 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:

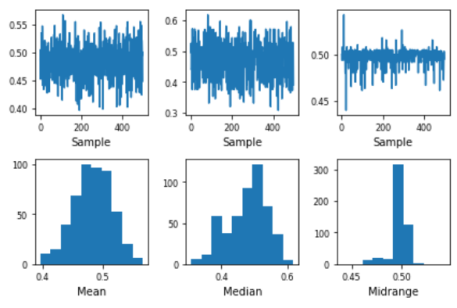
* which sub-sample had the lowest variance, or
* which sub-sample creates the narrowest confidence interval, etc.

**Implementation:**

* Python

|  |
| --- |
| import pandas as pd  import numpy as np    s = pd.Series(np.random.uniform(size=500))  pd.plotting.bootstrap\_plot(s) |

**Output**



**Limitation**

1. The bootstrap plot gives an estimation of the required information from the population, not the exact values.
2. It is highly dependent on the dataset given. It fails to give good results when a lot of subsets have repeated samples.
3. The bootstrap plot becomes ineffective when we are obtaining information that is highly dependent on the tail values. **[As shown in Fig 1]**

**Advantages of bootstrap:**

* 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

**Disadvantages of bootstrap:**

* 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.

1. What are predictive models, and how do they work? What are descriptive types, and how do you use them? Examples of both types of models should be provided. Distinguish between these two forms of models.

# Difference Between Descriptive and Predictive Data Mining

## ****Descriptive mining:****

This term is basically used to produce correlation, cross-tabulation, frequency etc. These technologies are used to determine the similarities in the data and to find existing patterns. One more application of descriptive analysis is to develop the captivating subgroups in the major part of the data available. This analytics emphasis on the summarization and transformation of the data into meaningful information for reporting and monitoring.

Examples of descriptive data mining include clustering, association rule mining, and anomaly detection. Clustering involves grouping similar objects together, while association rule mining involves identifying relationships between different items in a dataset. Anomaly detection involves identifying unusual patterns or outliers in the data.

### ****Predictive Data Mining:****

The main goal of this mining is to say something about future results not of current behaviour. It uses the supervised learning functions which are used to predict the target value. The methods come under this type of mining category are called classification, time-series analysis and regression. Modelling of data is the necessity of the predictive analysis, and it works by utilizing a few variables of the present to predict the future not known data values for other variables.

Examples of predictive data mining include regression analysis, decision trees, and neural networks. Regression analysis involves predicting a continuous outcome variable based on one or more predictor variables. Decision trees involve building a tree-like model to make predictions based on a set of rules. Neural networks involve building a model based on the structure of the human brain to make predictions.

#### ****The main differences between descriptive and predictive data mining are:****

**Purpose:**Descriptive data mining is used to describe the data and identify patterns and relationships. Predictive data mining is used to make predictions about future events.

**Approach:** Descriptive data mining involves analyzing historical data to identify patterns and relationships. Predictive data mining involves using statistical models and machine learning algorithms to identify patterns and relationships that can be used to make predictions.

**Output:**Descriptive data mining produces summaries and visualizations of the data. Predictive data mining produces models that can be used to make predictions.

**Timeframe:** Descriptive data mining is focused on analyzing historical data. Predictive data mining is focused on making predictions about future events.

**Applications:** Descriptive data mining is used in applications such as market segmentation, customer profiling, and product recommendation. Predictive data mining is used in applications such as fraud detection, risk assessment, and demand forecasting.

**Difference Between Descriptive and Predictive Data Mining:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Comparison** | **Descriptive Data Mining** | **Predictive Data Mining** |
| **1.** | **Basic** | **It determines, what happened in the past by analyzing stored data.** | **It determines, what can happen in the future with the help past data analysis.** |
| **2.** | **Preciseness** | **It provides accurate data.** | **It produces results does not ensure accuracy.** |
| **3.** | **Practical analysis methods** | **Standard reporting, query/drill down and ad-hoc reporting.** | **Predictive modelling, forecasting, simulation and alerts.** |
| **4.** | **Require** | **It requires data aggregation and data mining** | **It requires statistics and forecasting methods** |
| **5.** | **Type of approach** | **Reactive approach** | **Proactive approach** |
| **6.** | **Describe** | **Describes the characteristics of the data in a target data set.** | **Carry out the induction over the current and past data so that predictions can be made.** |
| **7.** | **Methods(in general)** | * **what happened?** * **where exactly is the problem?** * **what is the frequency of the problem?** | * **what will happen next?** * **what is the outcome if these trends continue?** * **what actions are required to be taken?** |

### ****Conclusion:****

In conclusion, descriptive and predictive data mining are two important techniques for discovering patterns and trends in large datasets. Descriptive data mining is used to summarize and describe the data, while predictive data mining is used to make predictions about future events. Both techniques have their own advantages and applications, and the choice of technique depends on the specific problem and the nature of the data.

### Frequently Asked Questions:

Q: Can descriptive data mining be used to make predictions?

A: No, descriptive data mining is focused on describing and summarizing the data, and does not involve making predictions about future events.

Q: Can predictive data mining be used to describe the data?

A: Yes, predictive data mining involves analyzing the data to identify patterns and relationships that can be used to make predictions, which can also provide insights into the data.

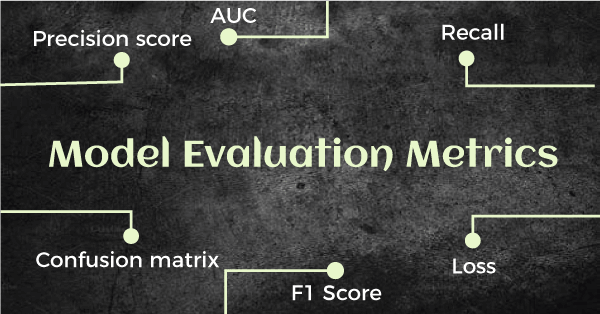
Q: What are some examples of applications that use predictive data mining?

A: Some examples of applications that use predictive data mining include credit scoring, insurance

1. Describe the method of assessing a classification model's efficiency in detail. Describe the various measurement parameters.

# Performance Metrics in Machine Learning

Evaluating the performance of a Machine learning model is one of the important steps while building an effective ML model. **To evaluate the performance or quality of the model, different metrics are used, and these metrics are known as performance metrics or evaluation metrics.** These performance metrics help us understand how well our model has performed for the given data. In this way, we can improve the model's performance by tuning the hyper-parameters. Each ML model aims to generalize well on unseen/new data, and performance metrics help determine how well the model generalizes on the new dataset.



In machine learning, each task or problem is divided into **classification** and **Regression**. Not all metrics can be used for all types of problems; hence, it is important to know and understand which metrics should be used. Different evaluation metrics are used for both Regression and Classification tasks. In this topic, we will discuss metrics used for classification and regression tasks.

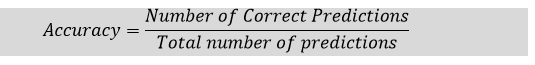
## 1. Performance Metrics for Classification

In a classification problem, the category or classes of data is identified based on training data. The model learns from the given dataset and then classifies the new data into classes or groups based on the training. It predicts class labels as the output, such as Yes or No, 0 or 1, Spam or Not Spam, etc. To evaluate the performance of a classification model, different metrics are used, and some of them are as follows:

* **Accuracy**
* **Confusion Matrix**
* **Precision**
* **Recall**
* **F-Score**
* **AUC(Area Under the Curve)-ROC**

### I. Accuracy

The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions. It can be formulated as:



To implement an accuracy metric, we can compare ground truth and predicted values in a loop, or we can also use the scikit-learn module for this.

Firstly, we need to import the accuracy\_score function of the scikit-learn library as follows:

1. from sklearn.metrics import accuracy\_score
2. Here, metrics is a class of sklearn.
3. Then we need to pass the ground truth and predicted values in the function to calculate the accuracy.
4. print(f'Accuracy Score is {accuracy\_score(y\_test,y\_hat)}')

Although it is simple to use and implement, it is suitable only for cases where an equal number of samples belong to each class.

**When to Use Accuracy?**

It is good to use the Accuracy metric when the target variable classes in data are approximately balanced. For example, if 60% of classes in a fruit image dataset are of Apple, 40% are Mango. In this case, if the model is asked to predict whether the image is of Apple or Mango, it will give a prediction with 97% of accuracy.

**When not to use Accuracy?**

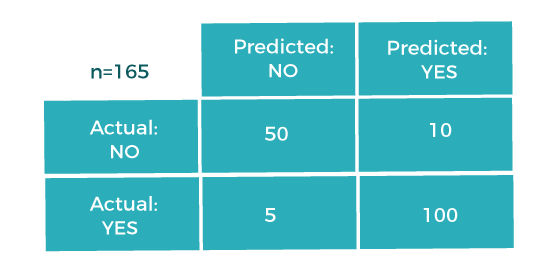
It is recommended not to use the Accuracy measure when the target variable majorly belongs to one class. For example, Suppose there is a model for a disease prediction in which, out of 100 people, only five people have a disease, and 95 people don't have one. In this case, if our model predicts every person with no disease (which means a bad prediction), the Accuracy measure will be 95%, which is not correct.

### II. Confusion Matrix

A confusion matrix is a tabular representation of prediction outcomes of any binary classifier, which is used to describe the performance of the classification model on a set of test data when true values are known.

The confusion matrix is simple to implement, but the terminologies used in this matrix might be confusing for beginners.

A typical confusion matrix for a binary classifier looks like the below image(However, it can be extended to use for classifiers with more than two classes).



We can determine the following from the above matrix:

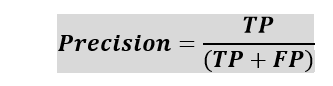
* In the matrix, columns are for the prediction values, and rows specify the Actual values. Here Actual and prediction give two possible classes, Yes or No. So, if we are predicting the presence of a disease in a patient, the Prediction column with Yes means, Patient has the disease, and for NO, the Patient doesn't have the disease.
* In this example, the total number of predictions are 165, out of which 110 time predicted yes, whereas 55 times predicted No.
* However, in reality, 60 cases in which patients don't have the disease, whereas 105 cases in which patients have the disease.

In general, the table is divided into four terminologies, which are as follows:

1. **True Positive(TP):** In this case, the prediction outcome is true, and it is true in reality, also.
2. True Negative(TN): in this case, the prediction outcome is false, and it is false in reality, also.
3. False Positive(FP): In this case, prediction outcomes are true, but they are false in actuality.
4. False Negative(FN): In this case, predictions are false, and they are true in actuality.

### III. Precision

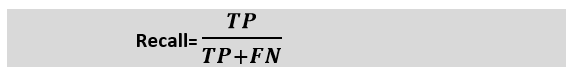
The precision metric is used to overcome the limitation of Accuracy. The precision determines the proportion of positive prediction that was actually correct. It can be calculated as the True Positive or predictions that are actually true to the total positive predictions (True Positive and False Positive).



### IV. Recall or Sensitivity

It is also similar to the Precision metric; however, it aims to calculate the proportion of actual positive that was identified incorrectly. It can be calculated as True Positive or predictions that are actually true to the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (true Positive and false negative).

The formula for calculating Recall is given below:



**When to use Precision and Recall?**

From the above definitions of Precision and Recall, we can say that recall determines the performance of a classifier with respect to a false negative, whereas precision gives information about the performance of a classifier with respect to a false positive.

So, if we want to minimize the false negative, then, Recall should be as near to 100%, and if we want to minimize the false positive, then precision should be close to 100% as possible.

In simple words, if we maximize precision, it will minimize the FP errors, and if we maximize recall, it will minimize the FN error.

### V. F-Scores

F-score or F1 Score is a metric to evaluate a binary classification model on the basis of predictions that are made for the positive class. It is calculated with the help of Precision and Recall. It is a type of single score that represents both Precision and Recall. So, **the F1 Score can be calculated as the harmonic mean of both precision and Recall, assigning equal weight to each of them.**

The formula for calculating the F1 score is given below:

Performance Metrics in Machine Learning

**When to use F-Score?**

As F-score make use of both precision and recall, so it should be used if both of them are important for evaluation, but one (precision or recall) is slightly more important to consider than the other. For example, when False negatives are comparatively more important than false positives, or vice versa.

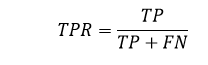
### VI. AUC-ROC

Sometimes we need to visualize the performance of the classification model on charts; then, we can use the AUC-ROC curve. It is one of the popular and important metrics for evaluating the performance of the classification model.

Firstly, let's understand ROC (Receiver Operating Characteristic curve) curve. **ROC represents a graph to show the performance of a classification model at different threshold levels**. The curve is plotted between two parameters, which are:

* **True Positive Rate**
* **False Positive Rate**

TPR or true Positive rate is a synonym for Recall, hence can be calculated as:



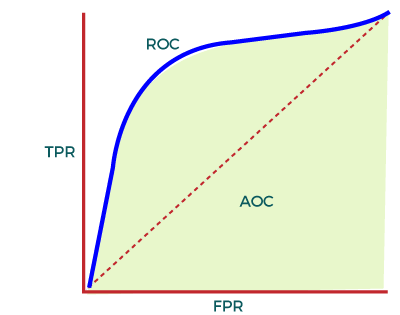
FPR or False Positive Rate can be calculated as:

Performance Metrics in Machine Learning

To calculate value at any point in a ROC curve, we can evaluate a logistic regression model multiple times with different classification thresholds, but this would not be much efficient. So, for this, one efficient method is used, which is known as AUC.

### AUC: Area Under the ROC curve

AUC is known for **Area Under the ROC curve**. As its name suggests, AUC calculates the two-dimensional area under the entire ROC curve, as shown below image:



AUC calculates the performance across all the thresholds and provides an aggregate measure. The value of AUC ranges from 0 to 1. It means a model with 100% wrong prediction will have an AUC of 0.0, whereas models with 100% correct predictions will have an AUC of 1.0.

**When to Use AUC**

AUC should be used to measure how well the predictions are ranked rather than their absolute values. Moreover, it measures the quality of predictions of the model without considering the classification threshold.

**When not to use AUC**

As AUC is scale-invariant, which is not always desirable, and we need calibrating probability outputs, then AUC is not preferable.

Further, AUC is not a useful metric when there are wide disparities in the cost of false negatives vs. false positives, and it is difficult to minimize one type of classification error.

## 2. Performance Metrics for Regression

Regression is a supervised learning technique that aims to find the relationships between the dependent and independent variables. A predictive regression model predicts a numeric or discrete value. The metrics used for regression are different from the classification metrics. It means we cannot use the Accuracy metric (explained above) to evaluate a regression model; instead, the performance of a Regression model is reported as errors in the prediction. Following are the popular metrics that are used to evaluate the performance of Regression models.

* **Mean Absolute Error**
* **Mean Squared Error**
* **R2 Score**
* **Adjusted R2**

### I. Mean Absolute Error (MAE)

Mean Absolute Error or MAE is one of the simplest metrics, which measures the absolute difference between actual and predicted values, where absolute means taking a number as Positive.

To understand MAE, let's take an example of Linear Regression, where the model draws a best fit line between dependent and independent variables. To measure the MAE or error in prediction, we need to calculate the difference between actual values and predicted values. But in order to find the absolute error for the complete dataset, we need to find the mean absolute of the complete dataset.

The below formula is used to calculate MAE:

Performance Metrics in Machine Learning

Here,

Y is the Actual outcome, Y' is the predicted outcome, and N is the total number of data points.

MAE is much more robust for the outliers. One of the limitations of MAE is that it is not differentiable, so for this, we need to apply different optimizers such as Gradient Descent. However, to overcome this limitation, another metric can be used, which is Mean Squared Error or MSE.

### II. Mean Squared Error

Mean Squared error or MSE is one of the most suitable metrics for Regression evaluation. It measures the average of the Squared difference between predicted values and the actual value given by the model.

Since in MSE, errors are squared, therefore it only assumes non-negative values, and it is usually positive and non-zero.

Moreover, due to squared differences, it penalizes small errors also, and hence it leads to over-estimation of how bad the model is.

MSE is a much-preferred metric compared to other regression metrics as it is differentiable and hence optimized better.

The formula for calculating MSE is given below:

Performance Metrics in Machine Learning

Here,

Y is the Actual outcome, Y' is the predicted outcome, and N is the total number of data points.

### III. R Squared Score

R squared error is also known as Coefficient of Determination, which is another popular metric used for Regression model evaluation. The R-squared metric enables us to compare our model with a constant baseline to determine the performance of the model. To select the constant baseline, we need to take the mean of the data and draw the line at the mean.

The R squared score will always be less than or equal to 1 without concerning if the values are too large or small.

Performance Metrics in Machine Learning

### IV. Adjusted R Squared

Adjusted R squared, as the name suggests, is the improved version of R squared error. R square has a limitation of improvement of a score on increasing the terms, even though the model is not improving, and it may mislead the data scientists.

To overcome the issue of R square, adjusted R squared is used, which will always show a lower value than R². It is because it adjusts the values of increasing predictors and only shows improvement if there is a real improvement.

We can calculate the adjusted R squared as follows:

Performance Metrics in Machine Learning

Here,

n is the number of observations

k denotes the number of independent variables

and Ra2 denotes the adjusted R2

4.

i. In the sense of machine learning models, what is underfitting? What is the most common reason for underfitting?

## ****Underfitting in Machine Learning****

A [statistical model](https://www.geeksforgeeks.org/difference-between-statistical-model-and-machine-learning/) or a machine learning algorithm is said to have underfitting when a model is too simple to capture data complexities. It represents the inability of the model to learn the training data effectively result in poor performance both on the training and testing data. In simple terms, an underfit model’s are inaccurate, especially when applied to new, unseen examples. It mainly happens when we uses very simple model with overly simplified assumptions. To address underfitting problem of the model, we need to use more complex models, with enhanced feature representation, and less regularization.

**Note: The underfitting model has High bias and low variance.**

### ****Reasons for**** ****Underfitting****

1. The model is too simple, So it may be not capable to represent the complexities in the data.
2. The input features which is used to train the model is not the adequate representations of underlying factors influencing the target variable.
3. The size of the training dataset used is not enough.
4. Excessive regularization are used to prevent the overfitting, which constraint the model to capture the data well.
5. Features are not scaled.

### ****Techniques to Reduce Underfitting****

1. Increase model complexity.
2. Increase the number of features, performing [feature engineering](https://www.geeksforgeeks.org/what-is-feature-engineering/).
3. Remove noise from the data.
4. Increase the number of [epochs](https://www.geeksforgeeks.org/epoch-in-machine-learning/) or increase the duration of training to get better results.

ii. What does it mean to overfit? When is it going to happen?

## ****Overfitting in Machine Learning****

A [statistical model](https://www.geeksforgeeks.org/difference-between-statistical-model-and-machine-learning/) is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise. The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.

In a nutshell, [Overfitting](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/) is a problem where the evaluation of machine learning algorithms on training data is different from unseen data.

### Reasons for Overfitting:

1. High variance and low bias.
2. The model is too complex.
3. The size of the training data.

### ****Techniques to Reduce Overfitting****

1. Improving the quality of training data reduces overfitting by focusing on meaningful patterns, mitigate the risk of fitting the noise or irrelevant features.
2. Increase the training data can improve the model’s ability to generalize to unseen data and reduce the likelihood of overfitting.
3. Reduce model complexity.
4. [Early stopping](https://www.geeksforgeeks.org/regularization-by-early-stopping/) during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
5. [Ridge Regularization](https://www.geeksforgeeks.org/lasso-vs-ridge-vs-elastic-net-ml/) and [Lasso Regularization](https://www.geeksforgeeks.org/implementation-of-lasso-regression-from-scratch-using-python/).
6. Use [dropout](https://www.geeksforgeeks.org/dropout-in-neural-networks/) for [neural networks](https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/) to tackle overfitting.

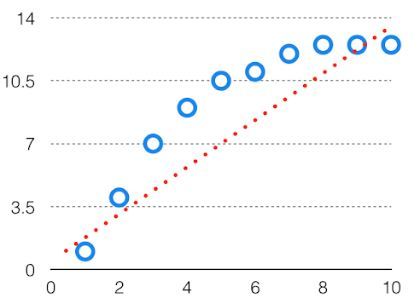
iii. In the sense of model fitting, explain the bias-variance trade-off

# Bias-Variance Trade Off – Machine Learning

It is important to understand prediction errors (bias and variance) when it comes to accuracy in any machine-learning algorithm. There is a tradeoff between a model’s ability to minimize bias and variance which is referred to as the best solution for selecting a value of [**Regularization**](https://www.geeksforgeeks.org/regularization-in-machine-learning/) constant. A proper understanding of these errors would help to avoid the overfitting and underfitting of a data set while training the algorithm.

## ****What is Bias?****

The bias is known as the difference between the prediction of the values by the [Machine Learning](https://www.geeksforgeeks.org/machine-learning/) model and the correct value. Being high in biasing gives a large error in training as well as testing data. It recommended that an algorithm should always be low-biased to avoid the problem of underfitting. By high bias, the data predicted is in a straight line format, thus not fitting accurately in the data in the data set. Such fitting is known as the **[Underfitting](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/) of Data**. This happens when the [hypothesis](https://www.geeksforgeeks.org/understanding-hypothesis-testing/) is too simple or linear in nature. Refer to the graph given below for an example of such a situation.

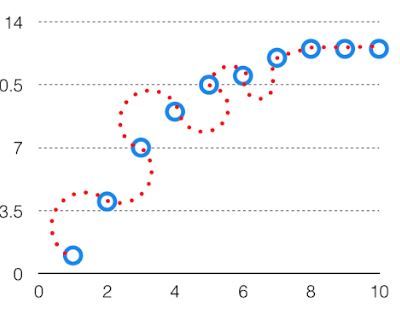


*High Bias in the Model*

In such a problem, a hypothesis looks like follows.

**What is Variance?**

The variability of model prediction for a given data point which tells us the spread of our data is called the variance of the model. The model with high variance has a very complex fit to the training data and thus is not able to fit accurately on the data which it hasn’t seen before. As a result, such models perform very well on training data but have high error rates on test data. When a model is high on variance, it is then said to as **Overfitting of Data**. Overfitting is fitting the training set accurately via complex curve and high order hypothesis but is not the solution as the error with unseen data is high. While training a data model variance should be kept low. The high variance data looks as follows.

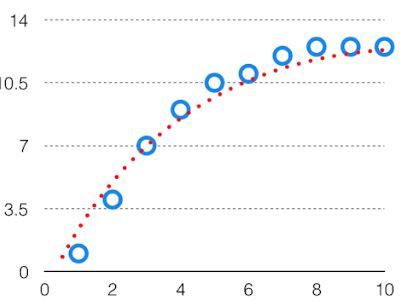


*High Variance in the Model*

In such a problem, a hypothesis looks like follows.

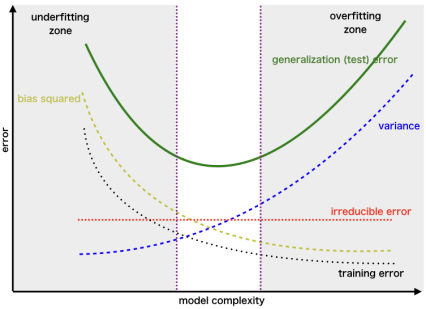
## ****Bias Variance Tradeoff****

If the algorithm is too simple (hypothesis with linear equation) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off. This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can’t be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like this.



We try to optimize the value of the total error for the model by using the [Bias Variance](https://www.geeksforgeeks.org/bias-vs-variance-in-machine-learning/) Tradeoff.

The best fit will be given by the hypothesis on the tradeoff point. The error to complexity graph to show trade-off is given as –



*Region for the Least Value of Total Error*

 This is referred to as the best point chosen for the training of the algorithm which gives low error in training as well as testing data.

.

1. Is it possible to boost the efficiency of a learning model? If so, please clarify how.

## 8 Methods to increase the accuracy of an ML Model

The model development cycle goes through various stages, starting from data collection to model building. But, before exploring the data to understand relationships (in variables), it’s always recommended to perform hypothesis generation. This step, often underrated in predictive modeling, is crucial for guiding your analysis effectively. By hypothesizing about potential relationships and patterns, you set the groundwork for a more targeted exploration. To know more about how to increase the accuracy of your machine learning model through effective hypothesis generation, refer to this link. It’s a key aspect that can significantly impact the success of your predictive modeling endeavors.

It is important that you spend time thinking about the given problem and gaining domain knowledge. So, how does it help? This practice usually helps in building better features later on, which are not biased by the data available in the dataset. This is a crucial step that usually improves a model’s accuracy.

At this stage, you are expected to apply structured thinking to the problem, i.e., a thinking process that takes into consideration all the possible aspects of a particular problem.

Let’s dig deeper now. Now we’ll check out the proven way how to increase accuracy of machine learning model:

1. Add More Data
2. Treat Missing and Outlier Values
3. Feature Engineering
4. Feature Selection
5. Multiple Algorithms
6. Algorithm Tuning
7. Ensemble Methods
8. Cross Validation

### Add More Data

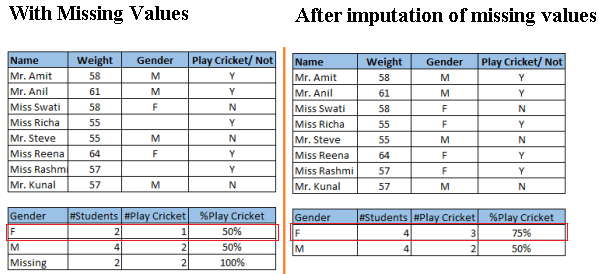
Having more data is always a good idea. It allows the “data to tell for itself” instead of relying on assumptions and weak correlations. Presence of more data results in better and more accurate machine-learning models.

I understand we don’t get an option to add more data. For example, we do not get a choice to increase the size of training data in data science competitions. But while working on a real-world company project, I suggest you ask for more data, if possible. This will reduce the pain of working on limited data sets.

### Treat Missing and Outlier Values

The unwanted presence of missing and outlier values in machine learning training data often reduces the accuracy of a trained model or leads to a biased model. It leads to inaccurate predictions. This is because we don’t analyze the behavior and relationship with other variables correctly. So, it is important to treat missing and outlier values well for a more reliable and naturally improved machine learning model.

Look at the below test data snapshot carefully. It shows that, in the presence of missing values, the chances of playing cricket by females are similar to males. But, if you look at the second table (after treatment of missing values based on the salutation “Miss”), we can see that females have higher chances of playing cricket compared to males.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/12/Data_Exploration_2_11.png)

Above, we saw the adverse effect of missing values on the accuracy of a trained model. Gladly, we have various methods to deal with missing and outlier values:

* **Missing:** In the case of continuous variables, you can impute the missing values with mean, median, or mode. For categorical variables, you can treat variables as a separate class. You can also build a model on the training dataset to predict the missing values. KNN imputation offers a great option to deal with missing values. To know more about these methods, refer to the article “[Methods to deal and treat missing values](https://www.analyticsvidhya.com/blog/2015/02/7-steps-data-exploration-preparation-building-model-part-2/)“.
* **Outlier:** You can delete the observations and perform transformations, binning, or imputation (same as missing values). Alternatively, you can also treat outlier values separately. You can refer article “[How to detect Outliers in your dataset and treat them?](https://www.analyticsvidhya.com/blog/2015/02/outliers-detection-treatment-dataset/)” to learn more about these methods.

### Feature Engineering

This step helps to extract more information from existing data. New information is extracted in terms of new features. These features may have a higher ability to explain the variance in the training data. Thus, giving improved model accuracy.

Feature engineering is highly influenced by hypothesis generation. Good hypotheses result in good features. That’s why I always suggest investing quality time in hypothesis generation. The feature engineering process can be divided into two steps:

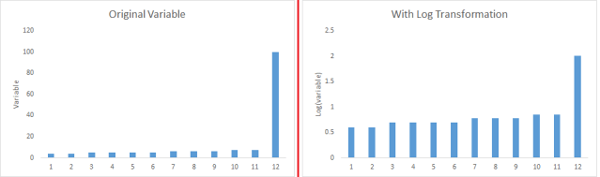
#### Feature Transformation

There are various scenarios where feature transformation is required:

Changing the scale of a variable from the original scale to a scale between zero and one is a common practice in machine learning, known as data normalization. For example, suppose a dataset includes variables measured in different units, such as meters, centimeters, and kilometers. Before applying any machine learning algorithm, it is essential to normalize these variables on the same scale to ensure fair and accurate comparisons. Normalization in machine learning contributes to better model performance and unbiased results across diverse variables.

Some algorithms work well with normally distributed data. Therefore, we must remove the skewness of variable(s). There are methods like a log, square root, or inverse of the values to remove skewness.

Sometimes, creating bins of numeric data works well since it handles the outlier values also. Numeric data can be made discrete by grouping values into bins. This is known as data discretization.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/12/Transformation_1.png)

#### Feature Creation

Deriving new variable(s) from existing variables is known as feature creation. It helps to unleash the hidden relationship of a data set. Let’s say we want to predict the number of transactions in a store based on transaction dates. Here transaction dates may not have a direct correlation with the number of transactions, but if we look at the day of the week, it may have a higher correlation.

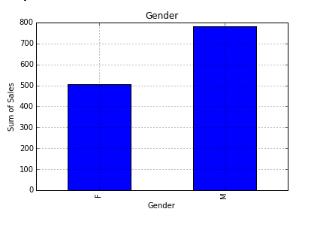
In this case, the information about the day of the week is hidden. We need to extract it to make the model accuracy better.Note that this might not be the case every time you create new features. This can also lead to a decrease in the accuracy or performance of the trained model. So every time creating a new feature, you must check the feature importance to see how that feature will affect the training process

### Feature Selection

Feature Selection is a process of finding out the best subset of attributes that better explains the relationship of independent variables with the target variable.

You can select the useful features based on various metrics like:

* **Domain Knowledge:**Based on domain experience, we select feature(s) which may have a higher impact on the target variable.
* **Visualization:**As the name suggests, it helps to visualize the relationship between variables, which makes your variable selection process easier.

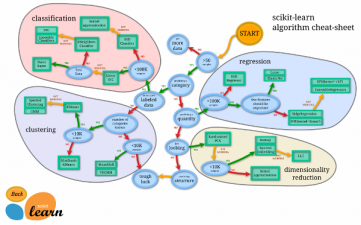
[](https://www.analyticsvidhya.com/wp-content/uploads/2015/12/box-plot.png)

* **Statistical Parameters:**We also consider the p-values, information values, and other statistical metrics to select the right features.
* **PCA:**It helps to represent training data into lower dimensional spaces but still characterizes the inherent relationships in the data. It is a type of dimensionality reduction technique. There are various methods to reduce training data’s dimensions (features), including factor analysis, low variance, higher correlation, backward/ forward feature selection, and others.

### Multiple Algorithms

There are many different algorithms in machine learning, but hitting the right machine learning algorithm is the ideal approach to how to increase accuracy of machine learning model. But, it is easier said than done.

This intuition comes with experience and incessant practice. Some algorithms are better suited to a particular type of data set than others. Hence, we should apply all relevant models and check the performance.



Source: Scikit-Learn cheat sheet

### Algorithm Tuning

We know that machine learning algorithms are driven by hyperparameters. These hyperparameters majorly influence the outcome of the learning process.

The objective of hyperparameter tuning is to find the optimum value for each hyperparameter how to increase accuracy of machine learning model. To tune these hyperparameters, you must have a good understanding of these meanings and their individual impact on the model. You can repeat this process with a number of well-performing models.

For example: In a random forest, we have various hyperparameters like max\_features, number\_trees, random\_state, oob\_score, and others. Intuitive optimization of these parameter values will result in better and more accurate models.

You can refer article “[Tuning the parameters of your Random Forest model](https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/)” to learn the impact of hyperparameter tuning in detail. Below is a random forest scikit learn algorithm with a list of all parameters:

RandomForestClassifier(n\_estimators=10, criterion='gini',

max\_depth=None,min\_samples\_split=2, min\_samples\_leaf=1,

min\_weight\_fraction\_leaf=0.0, max\_features='auto',

max\_leaf\_nodes=None,bootstrap=True, oob\_score=False, n\_jobs=1,

random\_state=None, verbose=0, warm\_start=False,class\_weight=None)

### Ensemble Methods

This is the most common approach that you will find majorly in winning solutions of Data science competitions. This technique simply combines the result of multiple weak models and produces better results. You can achieve by the following ways:

* **Bagging**(Bootstrap Aggregating)
* **Boosting**

To know more about these methods, you can refer article “[Introduction to ensemble learning](https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/)“.

It is always a better idea to implement ensemble methods to improve the accuracy of your model. There are two good reasons for this:

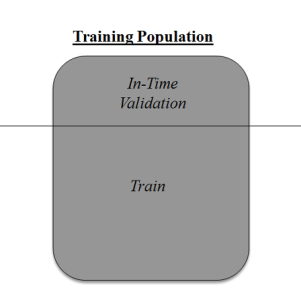
* They are generally more complex than traditional methods.
* The traditional methods give you a good base level from which you can improve and draw from to create your ensembles.

#### Caution!

Till here, we have seen methods that how to increase accuracy of machine learning model. But, it is not necessary that higher accuracy models always perform better (for unseen data points). Sometimes, the improvement in the model’s accuracy can be due to over-fitting too.

### Cross Validation

To find the right answer to this question, we must use the **cross-validation** technique. Cross Validation is one of the most important concepts in data modeling. It says to try to leave a sample on which you do not train the model and test the model on this sample before finalizing the model.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/12/validation.png)

This method helps us to achieve more generalized relationships. To know more about this cross-validation method, you should refer article “[Improve model performance using cross-validation](https://www.analyticsvidhya.com/blog/2015/11/improve-model-performance-cross-validation-in-python-r/)“.

6. How would you rate an unsupervised learning model's success? What are the most common success indicators for an unsupervised learning model?

**Metrics for Unsupervised Learning**

**Overview**

Metrics for unsupervised learning are utilized to degree the quality of a model's execution in unsupervised learning assignments. These measurements are regularly planned to degree the exactness of the clusters that are created by the model and/or the capacity of the model to precisely distinguish exceptions. Examples of metrics for unsupervised learning include homogeneity, completeness, V-measure, silhouette coefficient, and Davies–Bouldin index. Additionally, the results of unsupervised learning can be evaluated based on their utility and usability in terms of how well the model is able to identify meaningful patterns and relationships in the data.

**Metrics**

Measurements for unsupervised learning are utilized to assess the execution of clustering calculations, dimensionality reduction procedures, and other unsupervised learning strategies. There are a few common measurements utilized in unsupervised learning:

**Silhouette Coefficient:**

It takes values between -1 and 1, where a value near to 1 demonstrates that the information focuses inside a cluster are firmly stuffed, and the clusters are well-separated from other clusters. A value near to -1 demonstrates that the information focuses are misclassified or the clusters are overlapping. It is calculated by comparing the average distance between a data point and all other points in its cluster.

Silhouette Coefficient (SC)= (b-a)/max(a,b)

Where:

a = mean distance to other points in the same cluster

b = mean distance to other points in the next nearest cluster

**Example**: Let's say we have three clusters and three data points. The silhouette coefficient for each point can be calculated as follows:

**Point A:**

a = mean distance to other points in the same cluster (cluster 1) = 0.2

b = mean distance to other points in the next nearest cluster (cluster 2) = 0.7 SC = (0.7 - 0.2)/max(0.2, 0.7) = 0.78

**Point B:**

a = mean distance to other points in the same cluster (cluster 2) = 0.5

b = mean distance to other points in the next nearest cluster (cluster 3) = 0.6 SC = (0.6 - 0.5)/max(0.5, 0.6) = 0.17

**Point C:**

a = mean distance to other points in the same cluster (cluster 3) = 0.4

b = mean distance to other points in the next nearest cluster (cluster 1) = 0.9

SC = (0.9 - 0.4)/max(0.4, 0.9) = 0.67

**Calinski-Harabasz Index:**

The Calinski-Harabasz index measures the ratio of the between-cluster variance to the within-cluster variance. It takes higher values for clusters that are well-separated and dense.

Illustration: Consider a dataset comprising of two clusters, A and B. Cluster A comprises of 50 points and Cluster B comprises of 30 points, with their respective centroids found at (3, 5) and (7, 13), respectively. The within-cluster fluctuation for Cluster A is 0.2 and for Cluster B is 0.4. The between-cluster change is 0.6.

**Formula**: The Calinski-Harabasz index is calculated as follows:

Calinski-Harabasz index = (Between-cluster variance) / (Within-cluster variance)

= 0.6 / (0.2 + 0.4) = 1.5

Using the Calinski-Harabasz index, we can conclude that the two clusters (A and B) are relatively well-separated and dense, as the index value is greater than 1.

**Davies-Bouldin Index:**

The Davies-Bouldin index measures the average similarity between each cluster and its most similar cluster, relative to the average distance between each cluster and its most dissimilar cluster. It takes lower values for clusters that are well-separated and dense. The formula for the DBI is as follows:

DBI = (1/K) \* Σmax(sim(c\_i, c\_j))

Where K is the number of clusters, c\_i and c\_j are two different clusters, and sim is the similarity function (usually the ratio of the sum of intra-cluster distances to the distance between the two clusters).

For example, if we have three clusters A, B, and C, we can calculate the DBI as follows:

DBI = (1/3) \* (max(sim(A, B)) + max(sim(A, C)) + max(sim(B, C)))

For example, if the sum of intra-cluster distances for clusters A, B, and C are 10, 20, and 30 respectively, and the distance between clusters A and B is 8, A and C is 12, and B and C is 10, then the DBI would be:

DBI = (1/3) \* (max(8/10, 20/20) + max(8/10, 30/12) + max(20/20, 30/10)) = (1/3) \* (8/10 + 30/12 + 20/20) = (1/3) \* (2 + 2.5 + 1) = (1/3) \* 5.5 = 1.83

**Adjusted Rand Index:**

The adjusted rand index measures the similarity between the true labels and the predicted labels, taking into account chance agreements. It takes values between -1 and 1, where a value close to 1 indicates that the predicted labels are identical to the true labels.

The formula for the ARI is:

ARI = (Σi,j[aij - (ai.)(aj.)/n2 - (Σi(ai.)2/n2)(Σj(aj.)2/n2)]) / (0.5[Σi(ai.)2/n2 + Σj(aj.)2/n2])

where aij is the number of components within the crossing point of clusters i and j, ai. is the whole of components in cluster i, and n is the entire number of elements within the information set.

For illustration, on the off chance that you've got a information set of 100 components and two segments of the same information set, one with 50 components in cluster A and 50 components in cluster B, and the other with 60 components in cluster A and 40 components in cluster B, the ARI would be calculated as follows:

ARI = (5060 - (5050 + 5040)/100100 - (5050/100100)(4040/100100)) / (0.5[5050/100100 + 4040/100100])

ARI = (3000 - 2000/10000 - (2500/10000)(1600/10000)) / (0.5[2500/10000 + 1600/10000])

ARI = (1000 - 0.2) / (0.5\*(0.41))

ARI = 4.9 / 0.205

ARI = 24

**Mutual Information**:

The mutual information measures the amount of information shared between the true labels and the predicted labels. It takes values between 0 and 1, where a value close to 1 indicates that the predicted labels are identical to the true labels.

The formula for Mutual Information is as follows:

MI(X,Y) = ∑x∈X∑y∈Y p(x,y) log2 (p(x,y) / p(x)p(y))

For example, if we wanted to measure the mutual information between two variables X and Y, we could use the following:

Let X = Number of hours of sleep Let Y = Number of hours of studying

Then, MI(X,Y) = ∑x ∈ X ∑y ∈ Y p(x,y) log2 (p(x,y) / p(x)p(y))

**Conclusion**

It is important to note that different metrics may be appropriate for different types of unsupervised learning tasks, and the choice of metric should be made based on the specific requirements of the problem at hand.

**Key takeaways**

1. Use clustering metrics such as Silhouette Coefficient and Davies-Bouldin index to assess the quality and performance of clusters.
2. Use Adjusted Rand Index and Mutual Information to measure the similarity between two clusters.
3. Use Root Mean Square Error to measure the accuracy of the clustering.
4. Use perplexity and log-likelihood scores to measure the quality of a generative model.
5. Use precision-recall curves to measure the performance of a clustering algorithm.
6. Use silhouette scores to assess the degree of separation between clusters.
7. Use silhouette scores to assess the degree of overlap between clusters.
8. Use Average Linkage, Single Linkage, and Complete Linkage algorithms to measure the similarity between clusters.
9. Is it possible to use a classification model for numerical data or a regression model for categorical data with a classification model? Explain your answer.

# Classification vs Regression in Machine Learning

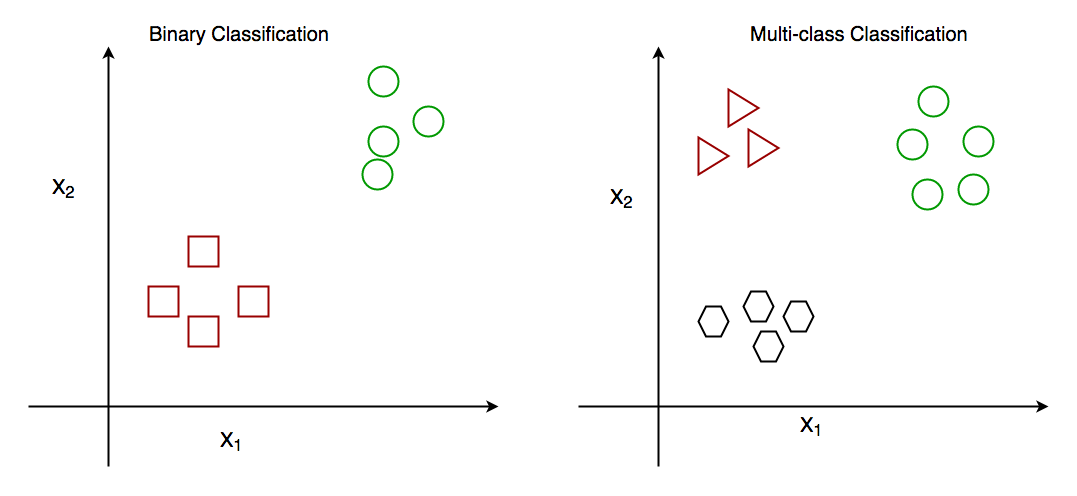
Classification and Regression are two major prediction problems that are usually dealt with in [Data Mining](https://www.geeksforgeeks.org/data-mining/) and [Machine Learning](https://www.geeksforgeeks.org/machine-learning/). We are going to deal with both Classification and Regression and we will also see differences between them in this article.

## Classification Algorithms

[**Classification**](https://www.geeksforgeeks.org/getting-started-with-classification/) is the process of finding or discovering a model or function that helps in separating the data into multiple categorical classes i.e. discrete values. In classification, data is categorized under different labels according to some parameters given in the input and then the labels are predicted for the data.

* In a classification task, we are supposed to predict discrete target variables(class labels) using independent features.
* In the classification task, we are supposed to find a [decision boundary](https://www.geeksforgeeks.org/ml-decision-function/) that can separate the different classes in the target variable.

The derived mapping function could be demonstrated in the form of “IF-THEN” rules. The classification process deals with problems where the data can be divided into binary or multiple discrete labels. Let’s take an example, suppose we want to predict the possibility of the winning of a match by Team A on the basis of some parameters recorded earlier. Then there would be two labels Yes and No.



*Binary Classification and Multiclass Classification*

### Types of Classification Algorithms

There are different types of State of the art classification algorithms that have been developed over time to give the best results for classification tasks by employing techniques like [bagging](https://www.geeksforgeeks.org/ml-bagging-classifier/) and [boosting](https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/).

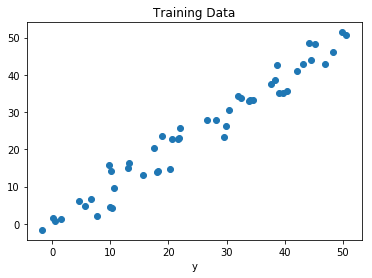
* [Decision Tree](https://www.geeksforgeeks.org/decision-tree/)
* [Random Forest Classifier](https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/)
* [K – Nearest Neighbors](https://www.geeksforgeeks.org/k-nearest-neighbours/)
* [Support Vector Machine](https://www.geeksforgeeks.org/support-vector-machine-algorithm/)

## Regression Algorithms

[**Regression**](https://www.geeksforgeeks.org/ml-linear-regression/) is the process of finding a model or function for distinguishing the data into continuous real values instead of using classes or discrete values. It can also identify the distribution movement depending on the historical data. Because a regression predictive model predicts a quantity, therefore, the skill of the model must be reported as an error in those predictions.

* In a regression task, we are supposed to predict a continuous target variable using independent features.
* In the regression tasks, we are faced with generally two types of problems linear and non-linear regression.

Let’s take a similar example in regression also, where we are finding the possibility of rain in some particular regions with the help of some parameters recorded earlier. Then there is a probability associated with the rain.



*Regression of Day vs Rainfall (in mm)*

### Types of Regression Algorithms

There are different types of State of the art regression algorithms that have been developed over time to give the best results for regression tasks by employing techniques like bagging and boosting.

* [Lasso Regression](https://www.geeksforgeeks.org/implementation-of-lasso-regression-from-scratch-using-python/)
* [Ridge Regression](https://www.geeksforgeeks.org/implementation-of-ridge-regression-from-scratch-using-python/)
* [XGBoost Regressor](https://www.geeksforgeeks.org/xgboost-for-regression/)
* [LGBM Regressor](https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/)

## Comparison between Classification and Regression

| **Classification** | **Regression** |
| --- | --- |
| In this problem statement, the target variables are discrete. | In this problem statement, the target variables are continuous. |
| Problems like [Spam Email Classification](https://www.geeksforgeeks.org/detecting-spam-emails-using-tensorflow-in-python/), [Disease prediction](https://www.geeksforgeeks.org/disease-prediction-using-machine-learning/) like problems are solved using Classification Algorithms. | Problems like [House Price Prediction](https://www.geeksforgeeks.org/house-price-prediction-using-machine-learning-in-python/), [Rainfall Prediction](https://www.geeksforgeeks.org/ml-rainfall-prediction-using-linear-regression/) like problems are solved using regression Algorithms. |
| In this algorithm, we try to find the best possible decision boundary which can separate the two classes with the maximum possible separation. | In this algorithm, we try to find the best-fit line which can represent the overall trend in the data. |
| [Evaluation metrics](https://www.geeksforgeeks.org/metrics-for-machine-learning-model/) like Precision, Recall, and F1-Score are used here to evaluate the performance of the classification algorithms. | Evaluation metrics like [Mean Squared Error,](https://www.geeksforgeeks.org/python-mean-squared-error/) [R2-Score](https://www.geeksforgeeks.org/ml-r-squared-in-regression-analysis/), and  [MAPE](https://www.geeksforgeeks.org/how-to-calculate-mape-in-python/) are used here to evaluate the performance of the regression algorithms. |
| Here we face the problems like [binary Classification](https://www.geeksforgeeks.org/getting-started-with-classification/) or [Multi-Class Classification](https://www.geeksforgeeks.org/multiclass-classification-using-scikit-learn/) problems. | Here we face the problems like [Linear Regression](https://www.geeksforgeeks.org/ml-linear-regression/) models as well as non-linear models. |
| Input Data are Independent variables and categorical dependent variable. | Input Data are Independent variables and continuous dependent variable. |
| The classification algorithm’s task mapping the input value of x with the discrete output variable of y. | The regression algorithm’s task is mapping input value (x) with continuous output variable (y). |
| Output is Categorical labels. | Output is Continuous numerical values. |
| Objective is to  Predict categorical/class labels. | Objective is to Predicting continuous numerical values. |
| Example use cases are Spam detection, image recognition, sentiment analysis | Example use cases are Stock price prediction, house price prediction, demand forecasting. |
| **Examples of classification algorithms are:**  Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (K-NN), Naive Bayes, Neural Networks, K-Means Clustering, Multi-layer Perceptron (MLP), etc. | **Examples of regression algorithms are:**  Linear Regression, Polynomial Regression, Ridge Regression, Lasso Regression, Support Vector Regression (SVR), Decision Trees for Regression, Random Forest Regression, K-Nearest Neighbors (K-NN) Regression, Neural Networks for Regression, etc. |

## When to Use Regression/Classification?

Classification trees are employed when there’s a need to categorize the dataset into distinct classes associated with the response variable. Often, these classes are binary, such as “Yes” or “No,” and they are mutually exclusive. While there are instances where there may be more than two classes, a modified version of the classification tree algorithm is used in those scenarios.

On the other hand, regression trees are utilized when dealing with continuous response variables. For instance, if the response variable represents continuous values like the price of an object or the temperature for the day, a regression tree is the appropriate choice.

There are situations where a blend of regression and classification approaches is necessary. For instance, ordinal regression comes into play when dealing with ranked or ordered categories, while multi-label classification is suitable for cases where data points can be associated with multiple classes at the same time.

1. Describe the predictive modeling method for numerical values. What distinguishes it from categorical predictive modeling?

# predictive modeling

* [**George Lawton**](https://www.techtarget.com/contributor/George-Lawton)
* [**Joseph M. Carew**](https://www.techtarget.com/contributor/Joseph-M-Carew)
* [**Ed Burns**](https://www.techtarget.com/contributor/Ed-Burns)

## What is predictive modeling?

Predictive modeling is a mathematical process used to predict future events or outcomes by analyzing patterns in a given set of input data. It is a crucial component of [predictive analytics](https://www.techtarget.com/searchbusinessanalytics/definition/predictive-analytics), a type of data analytics which uses current and historical data to forecast activity, behavior and trends.

Examples of [predictive modeling](https://www.techtarget.com/searchbusinessanalytics/tip/5-step-predictive-analytics-process-cycle) include estimating the quality of a sales lead, the likelihood of spam or the probability someone will click a link or buy a product. These capabilities are often baked into various business applications, so it is worth understanding the mechanics of predictive modeling to troubleshoot and improve performance.

Although predictive modeling implies a focus on forecasting the future, it can also predict outcomes (e.g., the probability a transaction is fraudulent). In this case, the event has already happened (fraud committed). The goal here is to predict whether future analysis will find the transaction is fraudulent. Predictive modeling can also forecast future requirements or facilitate what-if analysis.

"Predictive modeling is a form of data mining that analyzes historical data with the goal of identifying trends or patterns and then using those insights to predict future outcomes," explained Donncha Carroll a partner in the revenue growth practice of Axiom Consulting Partners. "Essentially, it asks the question, 'have I seen this before' followed by, 'what typically comes after this pattern.'"

### [What is predictive analytics? An enterprise guide](https://www.techtarget.com/searchbusinessanalytics/definition/predictive-analytics)

* Which also includes:
* [**Predictive analytics vs. machine learning**](https://www.techtarget.com/searchenterpriseai/feature/Machine-learning-and-predictive-analytics-work-better-together)
* [**7 top predictive analytics use cases: Enterprise examples**](https://www.techtarget.com/searchbusinessanalytics/feature/Top-5-predictive-analytics-use-cases-in-enterprises)
* [**Descriptive vs. prescriptive vs. predictive analytics explained**](https://www.techtarget.com/searchbusinessanalytics/tip/Descriptive-vs-prescriptive-vs-predictive-analytics-explained)

## Top types of predictive models

There are many ways of classifying predictive models and in practice multiple types of models may be combined for best results. The most salient distinction is between [unsupervised versus supervised models](https://www.techtarget.com/searchenterpriseai/feature/Comparing-supervised-vs-unsupervised-learning).

* Unsupervised models use traditional statistics to classify the data directly, using techniques like [logistic regression](https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression), time series analysis and decision trees.
* Supervised models use newer machine learning techniques such as neural networks to identify patterns buried in data that has already been labeled.

The biggest difference between these approaches is that with supervised models more care must be taken to properly label data sets upfront.

"The application of different types of models tends to be more domain-specific than industry-specific," said Scott Buchholz, government and public services CTO and emerging technology research director at Deloitte Consulting.

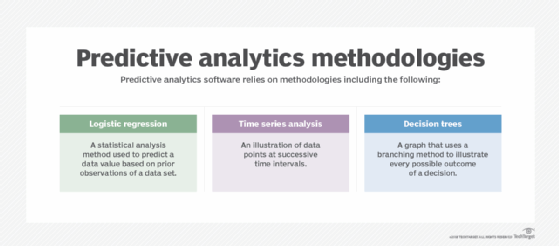
In certain cases, for example, standard statistical regression analysis may provide the best predictive power. In other cases, more sophisticated models are the right approach. For example, in a hospital, classic statistical techniques may be enough to identify key constraints for scheduling, but neural networks, a type of [deep learning](https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network), may be required to optimize patient assignment to doctors.

Once data scientists gather this sample data, they must select the right model. [Linear regressions](https://www.techtarget.com/searchenterpriseai/definition/linear-regression) are among the simplest types of predictive models. Linear models take two variables that are correlated -- one independent and the other dependent -- and plot one on the x-axis and one on the y-axis. The model applies a best fit line to the resulting data points. Data scientists can use this to predict future occurrences of the dependent variable.

Some of the most popular methods include the following:

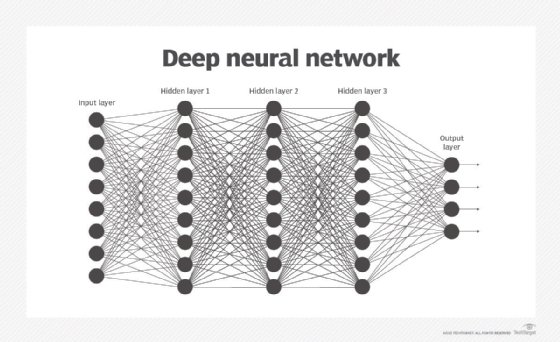
* **Decision trees.** Decision tree algorithms take data (mined, open source, internal) and graph it out in branches to display the possible outcomes of various decisions. Decision trees classify response variables and predict response variables based on past decisions, can be used with incomplete data sets and are easily explainable and accessible for novice data scientists.
* **Time series analysis.** This is a technique for the prediction of events through a sequence of time. You can predict future events by analyzing past trends and extrapolating from there.
* **Logistic regression.** This method is a statistical analysis method that aids in data preparation. As more data is brought in, the algorithm's ability to sort and classify it improves and therefore predictions can be made.
* **Neural networks.**This technique reviews large volumes of labeled data in search of correlations between variables in the data. Neural networks form the basis of many of today's examples of artificial intelligence (AI), including [image recognition](https://www.techtarget.com/whatis/definition/image-recognition), smart assistants and natural language generation.

The most complex area of predictive modeling is the [neural network](https://www.techtarget.com/searchnetworking/definition/neural-network). This type of machine learning model independently reviews large volumes of labeled data in search of correlations between variables in the data. It can detect even subtle correlations that only emerge after reviewing millions of data points. The algorithm can then make inferences about unlabeled data files that are similar in type to the data set it trained on.

Predictive modeling algorithms include logistic regression, time series analysis and decision trees.

## Common algorithms for predictive modeling

* **Random Forest.** This algorithm combines unrelated decision trees and uses classification and regression to organize and label vast amounts of data.
* **Gradient boosted model.** Similar to Random Forest, this algorithm uses several decision trees, but in this method, each tree corrects the flaws of the previous one and builds a more accurate picture.
* **K-Means.** This algorithm groups data points in a similar fashion as clustering models and is popular in devising personalized retail offers. It create personalized offers by seeking out similarities among large groups of customers.
* **Prophet.** A forecasting procedure, this algorithm is especially effective when dealing with capacity planning. This algorithm deals with time series data and is relatively flexible.

A neural network is a type of predictive model that independently reviews large volumes of labeled data in search of correlations between variables in the data.

## What are the uses of predictive modeling?

Predictive modeling is often associated with meteorology and weather forecasting, but [predictive models have many applications in business](https://www.techtarget.com/searchbusinessanalytics/feature/Top-5-predictive-analytics-use-cases-in-enterprises). Today's predictive analytics techniques can discover patterns in the data to identify upcoming risks and opportunities for an organization.

"Almost anywhere a smart human is regularly making a prediction in a historically data rich environment is a good use case for predicative analytics," Buchholz said. "After all, the model has no ego and won't get bored."

One of the most common uses of [predictive modeling is in online advertising and marketing](https://www.techtarget.com/searchbusinessanalytics/tip/Predictive-analytics-in-marketing-Achieving-success). Modelers use web surfers' historical data, to determine what kinds of products users might be interested in and what they are likely to click on.

[Bayesian spam filters](https://www.techtarget.com/whatis/definition/Bayesian-filter) use predictive modeling to identify the probability that a given message is spam.

In fraud detection, predictive modeling is used to identify outliers in a data set that point toward fraudulent activity. In customer relationship management, predictive modeling is used to target messaging to customers who are most likely to make a purchase.

Carroll said that predictive modeling is widely used in predictive maintenance, which has become a huge industry generating billions of dollars in revenue. One of the more notable examples can be found in the airline industry where engineers use IoT devices to remotely monitor performance of aircraft components like fuel pumps or jet engines.

These tools enable preemptive deployment of maintenance resources to increase equipment utilization and limit unexpected downtime. "These actions can meaningfully improve operational efficiency in a world that runs just in time where surprises can be very expensive," Caroll said.

Other areas where predictive models are used include the following:

* [capacity planning](https://searchenterprisewan.techtarget.com/definition/capacity-planning)
* [change management](https://searchcio-midmarket.techtarget.com/definition/change-management)
* disaster recovery
* engineering
* physical and digital security management
* city planning

## How to build a predictive model

Building a predictive model starts with identifying historical data that's representative of the outcome you are trying to predict.

"The model can infer outcomes from historical data but cannot predict what it has never seen before," Carroll said. Therefore, the volume and breadth of information used to train the model is critical to securing an accurate prediction for the future.

The next step is to identify ways to clean, transform and combine the raw data that leads to better predictions.

Skill is required in not only finding the appropriate set of raw data but also transforming it into data features that are most appropriate for a given model. For example, calculations of time-boxed weekly averages may be more useful and lead to better algorithms than real-time levels.

It is also important to weed out data that is coincidental or not relevant to a model. At best, the additional data will slow the model down, and at worst, it will lead to less accurate models.

This is [both an art and a science](https://www.techtarget.com/searchbusinessanalytics/opinion/Analytical-modeling-is-both-science-and-art). The art lies in cultivating a gut feeling for the meaning of things and intuiting the underlying causes. The science lies in methodically applying algorithms to consistently achieve reliable results, and then evaluating these algorithms over time. Just because a spam filter works on day one does not mean marketers will not tune their messages, making the filter less effective.

Analyzing representative portions of the available information -- [sampling](https://www.techtarget.com/searchbusinessanalytics/definition/data-sampling) -- can help speed development time on models and enable them to be deployed more quickly.

## Benefits of predictive modeling

Phil Cooper, group VP of products at Clari, a [RevOps](https://www.techtarget.com/searchcustomerexperience/definition/revenue-operations-RevOps) software startup, said some of the top [benefits of predictive modeling in business](https://www.techtarget.com/searchbusinessanalytics/tip/Benefits-of-predictive-analytics-for-businesses) include the following:

* **Prioritizing resources.** Predictive modeling is used to identify sales lead conversion and send the best leads to inside sales teams; predict whether a customer service case will be escalated and triage and route it appropriately; and predict whether a customer will pay their invoice on time and optimize accounts receivable workflows.
* **Improving profit margins.** Predictive modeling is used to forecast inventory, create pricing strategies, predict the number of customers and configure store layouts to maximize sales.
* **Optimizing marketing campaigns.** Predictive modeling is used to unearth new customer insights and predict behaviors based on inputs, allowing organizations to tailor marketing strategies, retain valuable customers and take advantage of cross-sell opportunities.
* **Reducing risk.** Predictive analytics can detect activities that are out of the ordinary such as fraudulent transactions, corporate spying or cyber attacks to reduce reaction time and negative consequences.

The techniques used in predictive modeling are probabilistic as opposed to deterministic. This means models generate probabilities of an outcome and include some uncertainty.

"This is a fundamental and inherent difference between [data modeling](https://www.techtarget.com/searchdatamanagement/definition/data-modeling) of historical facts versus predicting future events [based on historical data] and has implications for how this information is communicated to users," Cooper said. Understanding this difference is a critical necessity for transparency and explainability in how a prediction or recommendation was generated.

## Challenges of predictive modeling

Here are some of the [challenges related to predictive modeling](https://www.techtarget.com/searchbusinessanalytics/tip/Four-challenges-to-successful-predictive-analytics-models).

**Data preparation.** One of the most frequently overlooked challenges of predictive modeling is acquiring the correct amount of data and sorting out the right data to use when developing algorithms. By some estimates, data scientists spend about 80% of their time on this step. Data collection is important but limited in usefulness if this data is not properly managed and cleaned.

Once the data has been sorted, organizations must be careful to avoid overfitting. Over-testing on training data can result in a model that appears very accurate but has memorized the key points in the data set rather than learned how to generalize.

**Technical and cultural barriers.** While predictive modeling is often considered to be primarily a mathematical problem, users must plan for the technical and organizational barriers that might prevent them from getting the data they need. Often, systems that store useful data are not connected directly to centralized [data warehouses](https://searchsqlserver.techtarget.com/definition/data-warehouse). Also, some lines of business may feel that the data they manage is their asset, and they may not share it freely with data science teams.

**Choosing the right business case.** Another potential obstacle for predictive modeling initiatives is making sure projects address significant business challenges. Sometimes, data scientists discover correlations that seem interesting at the time and build algorithms to investigate the correlation further. However, just because they find something that is statistically significant does not mean it presents an insight the business can use. Predictive modeling initiatives need to have a solid foundation of business relevance.

**Bias.** "One of the more pressing problems everyone is talking about, but few have addressed effectively, is the challenge of bias," Carroll said. Bias is naturally introduced into the system through historical data since past outcomes reflect existing bias.

Nate Nichols, distinguished principal at Narrative Science, a natural language generation tools provider, is excited about the role that new explainable machine learning methods such as LIME or SHAP could play in addressing concerns about bias and promoting trust.

"People trust models more when they have some understanding of what the models are doing, and trust is paramount for predictive analytic capabilities," Nichols said. Being able to provide explanations for the predictions, he said, is a huge positive differentiator in the increasingly crowded field of predictive analytic products.

## Predictive modeling versus predictive analytics

Predictive modeling is but one aspect in the larger predictive analytics process cycle. This includes collecting, transforming, cleaning and modeling data using independent variables, and then reiterating if the model does not quite fit the problem to be addressed.

"Once data has been gathered, transformed and cleansed, then predictive modeling is performed on the data," said Terri Sage, chief technology officer at 1010data, an analytics consultancy.

Collecting data, transforming and cleaning are processes used for other types of analytic development.

"The difference with predictive analytics is the inclusion and discarding of variables during the iterative modeling process," Sage explained.

This will differ across various industries and use cases, as there will be diverse data used and different variables discovered during the modeling iterations.

For example, in healthcare, predictive models may ingest a tremendous amount of data pertaining to a patient and forecast a patient's response to certain treatments and prognosis. Data may include the patient's specific medical history, environment, social risk factors, genetics -- all which vary from person to person. [The use of predictive modeling in healthcare](https://www.techtarget.com/searchbusinessanalytics/tip/Predictive-analytics-in-healthcare-12-valuable-use-cases) marks a shift from treating patients based on averages to treating patients as individuals.

Similarly, with marketing analytics, predictive models might use data sets based on a consumer's salary, spending habits and demographics. Different data and modeling will be used for banking and insurance to help determine credit ratings and identify fraudulent activities.

## Predictive modeling tools

Before deploying a [predictive model tool](https://www.techtarget.com/searchbusinessanalytics/tip/6-top-predictive-analytics-tools), it is crucial for your organization to ask questions and sort out the following: Clarify who will be running the software, what the use case will be for these tools, what other tools will your predictive analytics be interacting with, as well as the budget.

Different tools have different data literacy requirements, are effective in different use cases, are best used with similar software and can be expensive. Once your organization has clarity on these issues, comparing tools becomes easier.

* **Sisense.** A business intelligence software aimed at a variety of companies that offers a range of business analytics features. This requires minimal IT background.
* **Oracle Crystal Ball.** A spreadsheet-based application focused on engineers, strategic planners and scientists across industries that can be used for predictive modeling, forecasting as well as simulation and optimization.
* **IBM SPSS Predictive Analytics Enterprise.** A business intelligence platform that supports open source integration and features descriptive and predictive analysis as well as data preparation.
* **SAS Advanced Analytics.** A program that offers algorithms that identify the likelihood of future outcomes and can be used for data mining, forecasting and econometrics.

## The future of predictive modeling

There are three key trends that will drive the future of data modeling.

1. First, data modeling capabilities are being baked into more business applications and citizen data science tools. These capabilities can provide the appropriate guardrails and templates for business users to work with predictive modeling.
2. Second, the tools and frameworks for low-code predictive modeling are making it easier for data science experts to quickly cleanse data, create models and vet the results.
3. Third, better tools are coming to automate many of the data engineering tasks required to push predictive models into production. Carroll predicts this will allow more organizations to shift from simply building models to deploying them in ways that deliver on their

9. The following data were collected when using a classification model to predict the malignancy of a group of patients' tumors:

i. Accurate estimates – 15 cancerous, 75 benign

ii. Wrong predictions – 3 cancerous, 7 benign

Determine the model's error rate, Kappa value, sensitivity, precision, and F-measure.

Cancer of unknown primary (CUP) site poses diagnostic challenges due to its elusive nature. Many cases of CUP manifest as pleural and peritoneal serous effusions. Leveraging cytological images from 57,220 cases at four tertiary hospitals, we developed a deep-learning method for tumor origin differentiation using cytological histology (TORCH) that can identify malignancy and predict tumor origin in both hydrothorax and ascites. We examined its performance on three internal (n = 12,799) and two external (n = 14,538) testing sets. In both internal and external testing sets, TORCH achieved area under the receiver operating curve values ranging from 0.953 to 0.991 for cancer diagnosis and 0.953 to 0.979 for tumor origin localization. TORCH accurately predicted primary tumor origins, with a top-1 accuracy of 82.6% and top-3 accuracy of 98.9%. Compared with results derived from pathologists, TORCH showed better prediction efficacy (1.677 versus 1.265, P < 0.001), enhancing junior pathologists’ diagnostic scores significantly (1.326 versus 1.101, P < 0.001). Patients with CUP whose initial treatment protocol was concordant with TORCH-predicted origins had better overall survival than those who were administrated discordant treatment (27 versus 17 months, P = 0.006). Our study underscores the potential of TORCH as a valuable ancillary tool in clinical practice, although further validation in randomized trials is warranted.

**Subject terms:**Cancer screening, Metastasis, Cancer of unknown primary

Developed on cytology images of hydrothorax and ascites from 57,220 cases at four hospitals, a deep-learning model shows high accuracy in tumor origin prediction and presents prognostic value when patient treatment is consistent with the cancer origin predicted by the model.

[Go to:](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/)

## Main

Cancers of unknown primary (CUP) site are a group of malignant diseases identified by histopathology as malignant metastases but whose origin cannot be identified by standard baseline diagnostic approaches. It is estimated that CUP accounts for 3–5% of all cancers diagnosed in humans[1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR1)–[4](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR4). Adenocarcinoma is the most common pathological type, followed by squamous and undifferentiated carcinoma[5](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR5),[6](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/" \l "CR6). Despite the employment of a variety of combined chemotherapies, the majority of patients have a very poor prognosis, with only 20% achieving a median survival of 10 months[7](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR7)–[10](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR10). CUP are often characterized by early dissemination, aggressive clinical course and multiple organ involvement. Immunohistochemistry is usually applied as a key means of predicting its probable origin; however, less than 30% of CUP cases can be pinpointed by cocktails of approximately 20 different immunostaining subunits[7](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR7),[11](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/" \l "CR11) and therefore CUP remain a thorny problem for clinicians. Accurate prediction of primary sites by pathologists and oncologists is a top priority for effective and personalized treatment.

Among patients newly diagnosed with CUP, a substantial portion present with pleural or peritoneal metastasis[7](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR7),[11](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/" \l "CR11),[12](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR12). The thoracic and abdominal serous cavities are locations where isolated tumor cells metastasize with high proclivity (Extended Data Fig. [​Fig.1).1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig6/)). Free tumor cells or implanted clusters found in pleural effusion or ascites are strong evidence of stage IV for some solid tumors[13](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR13)–[16](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR16). It has been reported that 7–20% of patients with respiratory or gastrointestinal tumors are diagnosed with pleural and peritoneal effusions, many of whom have synchronous peritoneal or pleural carcinomatosis[13](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR13)–[19](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR19). Previous studies revealed that serous effusions may develop without any history of cancer and present as the initial manifestation of cancer in 10% of patients with malignant effusions[20](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR20)–[23](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR23). Cytological examination by peritoneal or pleural fine-needle aspiration is usually used as a key method in the diagnosis of thoracoabdominal metastasis (Extended Data Fig. [​Fig.22](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig7/))[24](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR24)–[26](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR26). Most often, however, pathologists can visually distinguish adenocarcinoma from squamous carcinoma on cytology smears, but not the origin of the tumor cells[13](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR13),[23](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/" \l "CR23),[25](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR25). Therefore, precise cytological assessment may help in the appropriate management of patients with CUP and pleural or peritoneal metastasis, guide optimal therapeutic strategies, avoid unnecessary surgeries and further prolong overall survival[27](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR27)–[29](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR29).

[[An external file that holds a picture, illustration, etc.
Object name is 41591_2024_2915_Fig6_ESM.jpg](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig6/)](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig6/" \t "figure)

[Extended Data Fig. 1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig6/)

**A diagram illustrating tumor metastasis.**

Exemplified diagram shows the tumors from chest and abdominal organs have a high possibility of malignant hydrothorax and ascites.

[[An external file that holds a picture, illustration, etc.
Object name is 41591_2024_2915_Fig7_ESM.jpg](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig7_ESM.jpg)](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig7_ESM.jpg" \t "tileshopwindow)

[Extended Data Fig. 2](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig7/)

**Schematic diagram of cytological examination.**

Hydrothorax and ascites are punctured under the guidance of color Doppler ultrasound for cytological examination.

Computerized analysis based on deep convolutional neural networks has recently been increasingly applied as an auxiliary technique in the field of pathological diagnosis[30](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR30)–[32](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR32). Digital pathology has been applied to a variety of image-processing and image-classification tasks, including low-level object recognition and high-level disease prognosis or treatment-response prediction. Previous studies have reported the on-par performance of artificial intelligence (AI) models as compared with pathologists in the detection of breast cancer lymph node metastases, prediction of prostate cancer Gleason grading and interpretion of the likelihood of gastric cancer[33](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR33)–[35](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR35). Lu et al. also reported an AI model that showed potential benefits as a diagnostic assistive tool for CUP origin prediction using whole-slide images[36](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR36). However, these algorithms focused mainly on histological or whole-slide images; a deep-learning model that can interpret cytological imaging data to predict tumor origin is rarely reported[37](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR37). In routine clinical practice, histological and cytological pathologies have different application scenarios in terms of the tracking of tumor origin. Histological examination is used when specimens can be obtained via surgery or needle biopsy, these types of specimen providing richer diagnostic information. Cytology is mainly applicable for patients with late-stage cancer who cannot undergo surgery or tolerate needle biopsy[25](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR25),[38](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/" \l "CR38). In this scenario, specimens from pleural and peritoneal serous effusion are helpful in regard to localization of cancer origins due to their excellent accessibility[26](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR26),[39](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/" \l "CR39). However, sampling inadequacy (low cellular harvest), cellular degeneration or atypia and interexaminer variation in interpretation are major reasons for suboptimal diagnostic accuracy[25](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR25),[39](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/" \l "CR39),[40](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR40). Application of new techniques is required, such as AI auxiliary image analysis, to improve tumor detection capability. To the best of our knowledge, employment of AI in the prediction of cancer origin using cytological images from hydrothorax and ascites has not been investigated.

In this study we aimed to establish a diagnostic model to predict the broad cancer origins in patients with cancer and hydrothorax or ascites metastasis using cytological images. The performance of our AI system is examined and validated by large-scale cytological smear cases from several independent testing sets.

[Go to:](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/)

## Results

### Baseline characteristics of patients and image datasets

Between June 2010 and October 2023 we obtained a large dataset of 90,572 cytological smear images from 76,183 patients at four large institutions (Tianjin Medical University Cancer Institute and Hospital, Zhengzhou University First Hospital, Suzhou University First Hospital and Yantai Yuhuangding Hospital) as the training and testing sets (Table [​(Table1).1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/table/Tab1/)). We excluded 24,808 malignancy images lacking any clinical or pathological supporting evidence for the primary origins. A further 8,544 blank or poorly focused images were also excluded. The ultimate dataset consisted of 57,220 images from 43,688 patients (Extended Data Fig. [​Fig.3).3](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig8/)). The training set consisted of 29,883 images from 20,638 individuals covering 12 tumor subtypes or origins: 138, esophagus; 1,773, stomach; 20, intestine; 720, colon and rectum; 151, liver; 144, gallbladder; 357, pancreas; 321, uterus and vagina; 4,217, ovary and fallopian tube; 1,874, breast; 9,121, lung and upper respiratory tract; and 570, blood and lymphatic system. In addition to the 19,406 tumor images described above, 10,477 images of benign diseases were also included in the final training set. Similarly, three internal testing sets comprising 10,974 individuals (12,799 images) were obtained from the same four hospitals. Two additional external testing sets comprised 12,076 individuals (14,538 images) from Tianjin and Yantai hospitals (Fig. [​(Fig.1).1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig1/)). The tumor category of testing sets was broadly in line with that of the training set. Because one patient might have undergone more than one hydrothorax or ascites core needle biopsy for cytological analysis at various stages of disease development, more than one image may have been recorded. In this study, each image combined with its clinicopathological data was compiled as one case. Respiratory diseases accounted for the largest proportion (29.8%, n = 17,058) among malignant groups. Carcinoma amounted to 56.7% (n = 32,424) of overall hydrothorax and ascites cytological cases, among which adenocarcinoma comprised the largest category (47.2%, n = 27,006). The proportion of squamous cell carcinoma metastasizing to pleural effusion or ascites was only 0.6% (n = 346). In addition, there were 24,658 (82.5%) cases in the training set stratified as high certainty and 5,225 (17.5%) as low certainty. For the testing sets, 18,184 (66.5%) cases were stratified as high certainty and 9,153 (33.5%) as low certainty. With respect to images of malignancy, 6,066 of 19,406 (31.2%) cases in the training set and 4,256 of 16,702 (25.5%) cases in the testing sets also underwent examination by sediment paraffin immunohistochemical staining.

### Table 1

Baseline characteristics of training and testing sets

| **Parameter** | **Overall, n = 57,220 (%)** | **Training sets (n = 29,883)** | | | **Internal testing sets (n = 12,799)** | | | **External testing sets (n = 14,538)** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tianjin, n = 9,822 (%)** | **Zhengzhou, n = 14,586 (%)** | **Suzhou, n = 5,475 (%)** | **Tianjin, n = 4,186 (%)** | **Zhengzhou, n = 6,234 (%)** | **Suzhou, n = 2,379 (%)** | **Tianjin-Pc, n = 3 933 (%)** | **Yantai, n = 10,605 (%)** |
| Male sex | 25,822 (45.1) | 3,223 (32.8) | 7,353 (50.4) | 2,862 (52.3) | 1,369 (32.7) | 3,111 (49.9) | 1,235 (51.9) | 1,792 (45.6) | 4,877 (46.0) |
| Female sex | 31,398 (54.9) | 6,599 (67.2) | 7,233 (49.6) | 2,613 (47.7) | 2,817 (67.3) | 3,123 (50.1) | 1,144 (48.1) | 2,141 (54.4) | 5,728 (54.0) |
| Age, years (mean ± SD) | 59.13 ± 14.21 | 58.23 ± 11.47 | 57.17 ± 16.03 | 63.02 ± 14.40 | 58.25 ± 11.54 | 57.15 ± 15.94 | 63.57 ± 14.09 | 60.27 ± 12.58 | 60.73 ± 13.32 |
| Age ≤60 years | 28,079 (49.1) | 5,386 (54.8) | 7,702 (52.8) | 2,019 (36.9) | 2,284 (54.6) | 3,287 (52.7) | 854 (35.9) | 1,797 (45.7) | 4,750 (44.8) |
| Age >60 years | 29.141 (50.9) | 4,436 (45.2) | 6,884 (47.2) | 3,456 (63.1) | 1,902 (45.4) | 2,947 (47.3) | 1,525 (64.1) | 2,136 (54.3) | 5,855 (55.2) |
| Primary tumor site | | | | | | | | | |
| Digestive | 5,682 (9.9) | 1,504 (15.3) | 1,135 (7.8) | 664 (12.1) | 591 (14.1) | 544 (8.7) | 270 (11.3) | 315 (8.0) | 659 (6.2) |
| Female reproductive | 12,350 (21.6) | 3,901 (39.7) | 1,772 (12.1) | 739 (13.5) | 1,662 (39.7) | 774 (12.4) | 331 (13.9) | 888 (22.6) | 2,283 (21.5) |
| Respiratory | 17,058 (29.8) | 3,239 (33.0) | 3,742 (25.7) | 2,140 (39.1) | 1,433 (34.2) | 1,589 (25.5) | 958 (40.3) | 1,135 (28.9) | 2,822 (26.6) |
| Blood and lymphatic | 1,018 (1.8) | 73 (0.7) | 417 (2.9) | 80 (1.5) | 30 (0.7) | 214 (3.4) | 33 (1.4) | 63 (1.6) | 108 (1.0) |
| Benign | 21,112 (36.9) | 1,105 (11.3) | 7,520 (51.6) | 1,852 (33.8) | 470 (11.2) | 3,113 (49.9) | 787 (33.1) | 1,532 (39.0) | 4,733 (44.6) |
| Hydrothorax | 35,873 (62.7) | 5,751 (58.6) | 9,427 (64.6) | 3,803 (69.5) | 2,491 (59.5) | 4,031 (64.7) | 1,637 (68.8) | 2,364 (60.1) | 6,369 (60.1) |
| Ascites | 21,347 (37.3) | 4,071 (41.4) | 5,159 (35.4) | 1,672 (30.5) | 1,695 (40.5) | 2,203 (35.3) | 742 (31.2) | 1,569 (39.9) | 4,236 (39.9) |
| Carcinoma | 32,424 (56.7) | 7,944 (80.9) | 5,250 (36.0) | 3,203 (58.5) | 3,670 (87.7) | 2,900 (46.5) | 1,547 (65.0) | 2,335 (59.4) | 5,575 (52.6) |
| Adenocarcinoma | 27,006 (47.2) | 7,218 (73.5) | 4,279 (29.3) | 2,622 (47.9) | 3,022 (72.2) | 1,851 (29.7) | 1,129 (47.5) | 2,056 (52.3) | 4,829 (45.5) |
| Squamous carcinoma | 346 (0.6) | 50 (0.5) | 130 (0.9) | 30 (0.5) | 24 (0.6) | 60 (1.0) | 23 (1.0) | 7 (0.2) | 22 (0.2) |
| Other carcinoma  a | 1,518 (2.7) | 166 (1.7) | 294 (2.0) | 207 (3.8) | 53 (1.3) | 111 (1.8) | 88 (3.7) | 185 (4.7) | 414 (3.9) |
| Unclassified  b | 3,554 (6.2) | 510 (5.2) | 547 (3.8) | 344 (6.3) | 571 (13.6) | 878 (14.1) | 307 (12.9) | 87 (2.2) | 310 (2.9) |
| High-certainty cases | 42,912 (75.0) | 9,355 (95.2) | 10,708 (73.4) | 4,595 (83.9) | 3,992 (95.4) | 4,270 (68.5) | 1,981 (83.3) | 2,052 (52.2) | 5,959 (56.2) |
| Low-certainty cases | 14,308 (25.0) | 467 (4.8) | 3,878 (26.6) | 880 (16.1) | 194 (4.6) | 1,964 (31.5) | 398 (16.7) | 1,881 (47.8) | 4,646 (43.8) |

[Open in a separate window](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/table/Tab1/?report=objectonly)

aOther types consist mainly of sarcomatoid carcinoma, adenosquamous carcinoma, papillary carcinoma, large cell carcinoma, small cell carcinoma, transitional epithelial carcinoma, basal cell carcinoma and undifferentiated carcinoma.

bUnclassified carcinoma means that the specific type of cancer is unknown, the main reason being that the specimen was too small for staining by immunohistochemistry.

cTianjin-P, Tianjin external testing set enrolled prospectively.

[[An external file that holds a picture, illustration, etc.
Object name is 41591_2024_2915_Fig1_HTML.jpg](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig1_HTML.jpg)](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig1_HTML.jpg" \t "tileshopwindow)

[Fig. 1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig1/)

**Our proposed TORCH model framework.**

**a**, A total of 42,682 cases were sourced from three large tertiary referral institutions, 70% of which (n = 29,883) were used as training sets. Clinicopathological data were acquired from radiological imaging departments, medical records systems and pathological digital databases. **b**, During the diagnostic process, most images were magnified either ×200 or ×400. **c**, The deep-learning network, trained with cytological images, was aimed at dividing target images into five categories according to the highest predicted probability score. Classification results were further validated at four institutions, including three internal testing sets (n = 12,799) and two external testing sets (n = 14,538). N represents the N-th image tile.

[[An external file that holds a picture, illustration, etc.
Object name is 41591_2024_2915_Fig8_ESM.jpg](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig8_ESM.jpg)](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig8_ESM.jpg" \t "tileshopwindow)

[Extended Data Fig. 3](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig8/)

**The flowchart exhibiting the procedures to develop and evaluate TORCH model.**

**a**, Model development procedure consisted of feature extraction, real clinical data taxonomy and model iteration. **b**, Evaluation of TORCH on three internal and two external testing sets. **c**, Performance comparison between TORCH and four pathologists on randomly selected cases.

### Performance of TORCH on prediction of tumor origin

We developed TORCH by training four different deep neural networks on three different types of input, giving rise to 12 different models ([Methods](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#Sec12)). The classification results of each individual model are shown in Supplementary Figs. [1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM1)–[4](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM1) and Supplementary Tables [1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM3)–[4](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM3). We subsequently performed model ensembling to integrate these models ([Methods](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#Sec12)). The results showed that TORCH provides relatively reliable generalization and interoperability. On the five testing sets (n = 27,337), TORCH achieved an overall microaveraged one-versus-rest area under the receiver operating characteristic (AUROC) value of 0.969 (95% confidence interval (CI) 0.967–0.970). On the three internal testing sets, microaveraged one-versus-rest AUROC values were 0.953 (CI 0.949–0.958) for the Tianjin dataset, 0.962 (CI 0.960–0.965) for the Zhengzhou dataset and 0.979 (CI 0.976–0.983) for the Suzhou dataset (Fig. [​(Fig.2).2](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig2/)). On the two external testing sets, microaveraged one-versus-rest AUROC values were 0.958 (CI 0.954–0.962) and 0.978 (CI 0.977–0.980) for the Tianjin-P and Yantai datasets, respectively. In terms of identification of cancer-positive cases, TORCH achieved an AUROC value of 0.974 (CI 0.972–0.976), accuracy of 92.6% (CI 92.2–92.9%), sensitivity of 92.8% (CI 92.3–93.2%) and specificity of 92.4% (CI 92.0–92.8%) (Extended Data Table [​Table1).1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/table/Tab2/)). In terms of tumor origin localization in the female reproductive system group, TORCH achieved an AUROC value of 0.960 (CI 0.958–0.962), accuracy of 88.1% (CI 87.7–88.5%), sensitivity of 92.5% (CI 91.8–93.2%) and specificity of 86.9% (CI 86.4–87.3%), an enhanced performance compared with that for the other systems. In addition, the effectiveness of this model was stable in that it achieved similar results among the five testing sets. Detailed classification metrics of the five categories are provided in Extended Data Table [​Table22](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/table/Tab3/) and Supplementary Table [5](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM1). The model prediction results of 27,337 cases are shown in Supplementary Table [6](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM3).

[[An external file that holds a picture, illustration, etc.
Object name is 41591_2024_2915_Fig2_HTML.jpg](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig2_HTML.jpg)](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig2_HTML.jpg" \t "tileshopwindow)

[Fig. 2](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig2/)

**Classification performance of the TORCH model.**

**a**, The confusion matrix, including precision and recall, is plotted for prediction of isolated tumor cell origin on the overall five testing sets (n = 27,337). Microaveraged one-versus-rest ROC curves for the five categories (red curves). Top-n model (n = 1, 2, 3) accuracy for tumor origin classification. **b**–**f**, Five ROC curves for the auxiliary binary task of prediction of malignancy or benignity and prediction of four tumor categories (green curves). **b**, Tianjin testing set. **c**, Zhengzhou testing set. **d**, Suzhou testing set. **e**, Tianjin-P testing set. **f**, Yantai testing set. AUC, area under the curve.

### Extended Data Table 1

Classification performance of TORCH model on malignant versus benign diseases

[[An external file that holds a picture, illustration, etc.
Object name is 41591_2024_2915_Tab1_ESM.jpg](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Tab1_ESM.jpg)](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Tab1_ESM.jpg" \t "tileshopwindow)

Classification performance of TORCH model on malignant versus benign diseases

Classification performance of TORCH model on malignant versus benign diseases.

### Extended Data Table 2

Classification performance of TORCH model on five categories

[[An external file that holds a picture, illustration, etc.
Object name is 41591_2024_2915_Tab2_ESM.jpg](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Tab2_ESM.jpg)](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Tab2_ESM.jpg" \t "tileshopwindow)

Classification performance of TORCH model on five categories

Classification performance of TORCH model on five categories.

TORCH achieved a top-1 accuracy of 82.6%, top-2 accuracy of 95.9% and top-3 accuracy of 98.9% when combining these five testing sets. These top-n accuracies fluctuated within a narrow range among the five testing sets (Fig. [​(Fig.2).2](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig2/)). On the Tianjin internal testing set (n = 4,186), the top-n accuracies achieved by TORCH were 76.3, 95.7 and 99.1%, respectively; on the Zhengzhou testing set (n = 6,234), these were 80.8, 94.7 and 98.6%, respectively; and on the Suzhou testing set (n = 2,379), these were 87.4, 96.8 and 99.3%, respectively. With respect to stratification by specimen sampling site, TORCH achieved higher microaveraged one-versus-rest AUROC (0.970 (CI 0.969–0.972)) in the hydrothorax group than in the ascites group (0.966 (CI 0.964–0.969; P < 0.001); Supplementary Fig. [5](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM1) and Supplementary Table [7](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM3)). Among the five categories, TORCH achieved higher AUROC values in ascites than in hydrothorax for the digestive (0.892 versus 0.775, P < 0.001) and female reproductive systems (0.951 versus 0.945, P = 0.012) and lower AUROC values for the respiratory system (0.808 versus 0.929, P < 0.001). No significant differences were observed for benign diseases (0.972 versus 0.975, P = 0.068) or the blood and lymphatic system (0.967 versus 0.951, P = 0.122) in ascites versus hydrothorax. In addition, when solid tumors were divided into carcinoma and noncarcinoma, we observed that TORCH achieved comparable AUROC values in both the carcinoma group (0.938 (CI 0.936–0.940)) and the noncarcinoma group (0.939 (CI 0.921–0.958); P = 0.244). Within the carcinoma group, TORCH exhibited slightly better performance for the adenocarcinoma group versus the nonadenocarcinoma group (AUROC, 0.942 (CI 0.939–0.944) versus 0.925 (CI 0.919–0.931) (P = 0.002)).

To explore TORCH further we examined its prediction efficiency on both high- and low-certainty cases. TORCH achieved comparable microaveraged one-versus-rest AUROC values in the low-certainty group compared with the high-certainty group (0.964 (CI 0.961–0.966) versus 0.971 (CI 0.969–0.972), P = 0.106; Extended Data Fig. [​Fig.4).4](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig9/)). Meanwhile, no significant difference in terms of classification metrics was observed between the two subgroups. Classification metrics including accuracy, sensitivity, specificity, precision and negative predictive value are shown in Supplementary Table [8](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM3).

[[An external file that holds a picture, illustration, etc.
Object name is 41591_2024_2915_Fig9_ESM.jpg](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig9_ESM.jpg)](https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=11108774_41591_2024_2915_Fig9_ESM.jpg" \t "tileshopwindow)

[Extended Data Fig. 4](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig9/)

**Classification performance of TORCH model on high-certainty cases and low-certainty cases respectively.**

Overall micro-averaged one-versus-rest auroc is similar for cases in the low-certainty group (**b**) compared with high-certainty group (**a**) [0.964 (0.961–0.966) versus 0.971 (0.969–0.972) (P = 0.106)].

To further verify the generalization and reliability of TORCH, we enrolled 4,520 consecutive cases from Tianjin Cancer Hospital (the Tianjin-P dataset) and 12,467 from Yantai Hospital (the Yantai dataset) as fully unseen external testing sets. These images were collected from pathological databases without exclusion of any cases. The Tianjin-P and Yantai datasets included 587 and 1,862 uncertainty cases, respectively. We observed that TORCH achieved top-1/2/3 accuracy of 79.3, 94.4 and 98.3%, respectively, on the Tianjin-P dataset without uncertainty cases (n = 3,933) and 86.3, 97.1 and 99.2%, respectively, on the Yantai dataset without uncertainty cases (n = 10,605). The lower-bound top-1 accuracy of TORCH was estimated to be 70.2% on the Tianjin-P dataset and 75.1% on the Yantai dataset by assuming that all predictions made by TORCH for these uncertainty cases were erroneous. The upper-bound top-1 accuracy of TORCH was estimated to be 81.7% on the Tianjin-P dataset and 88.1% on the Yantai dataset by assuming that all predictions made by TORCH for these uncertainty cases were correct.

### Performance of TORCH versus pathologists

We asked two junior and two senior practicing pathologists to manually interpret 495 cytological images that comprised 333 malignant cases and 162 benign cases, with subsequent comparison with predictions made by TORCH. We observed that top-1 accuracies were 42.6% (95% CI 38.2–46.9%) and 44.0% (95% CI 39.4–47.9%) for the two junior pathologists and 69.7% (95% CI 66.3–73.5%) and 57.0% (95% CI 52.9–61.2%) for the two senior pathologists. Notably, TORCH achieved a top-1 accuracy of 78.8% (95% CI 75.4–82.0%), which was significantly higher than that for the four pathologists (permutation test, all P < 0.001). When stratified by the five categories, TORCH outperformed pathologists with respect to accuracy (mean 0.896 versus 0.813; P = 0.038), sensitivity (mean 0.880 versus 0.485; P < 0.001) and precision (mean 0.634 versus 0.486; P < 0.001; Extended Data Table [​Table3).3](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/table/Tab4/)). TORCH also achieved marginally higher specificity compared with this group of pathologists, although the difference did not reach statistical significance (mean 89.4% versus 87.8%; P = 0.333). Receiver operating characteristic (ROC) curves of TORCH for the five categories of these 495 cases are provided in Supplementary Fig. [6](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM1). TORCH achieved significantly higher diagnostic scores compared with the pathologists (1.677 (95% CI 1.647–1.706) versus 1.265 (95% CI 1.227–1.302), P < 0.001). The senior pathologists also achieved higher diagnostic scores compared with the junior pathologists (1.428 (95% CI 1.378–1.479) versus 1.101 (95% CI 1.047–1.155), P < 0.001; Supplementary Table [9](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM3)). Inter-rater agreement rate for the four pathologists was 24.6% (122 of 495, Fleiss’ kappa 0.365, two-sided z-test, P < 0.001). Although inter-rater agreement rate was statistically significant, it was still relatively low among the pathologists and could be considered to be in fair agreement according to Landis and Koch[41](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#CR41). This suggested that interpretation of cytological images for assessment of tumor origin is subject to substantial variability. In addition, the senior pathologists achieved significantly higher performance compared with their junior counterparts in terms of both accuracy (0.853 versus 0.773, P = 0.014) and precision (0.594 versus 0.381, P = 0.001; Supplementary Table [9](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM3)). In addition, both TORCH and the senior pathologists recorded higher sensitivity than the junior pathologists in differentiation of benign diseases from malignant tumors (Fig. [​(Fig.3).3](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/figure/Fig3/)). The performances of both senior and junior pathologists are shown in Supplementary Tables [9](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM3)–[12](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11108774/#MOESM3) and Supplementary

10. Make quick notes on:

1. The process of holding out

**Introduction of Holdout Method**

**Holdout Method** is the simplest sort of method to evaluate a classifier. In this method, the data set (a collection of data items or examples) is separated into two sets, called the **Training set and Test set**.

A classifier performs function of assigning data items in a given collection to a target category or class.

**Example –**  
E-mails in our inbox being classified into spam and non-spam.

Classifier should be evaluated to find out, it’s accuracy, error rate, and error estimates. It can be done using various methods. One of most primitive methods in evaluation of classifier is **‘Holdout Method’**.

In the holdout method, data set is partitioned, such that – maximum data belongs to training set and remaining data belongs to test set.

**Example –**  
If there are 20 data items present, 12 are placed in training set and remaining 8 are placed in test set.

* After partitioning data set into two sets, training set is used to build a model/classifier.
* After construction of classifier, we use data items in test set, to test accuracy, error rate and error estimate of model/classifier.

However, it is vital to remember two statements with regard to holdout method. These are :

If maximum possible data items are placed in training set for construction of model/classifier, classifier’s error rates and estimates would be very low and accuracy would be high. This is sign of a good classifier/model.

**Example –**  
A student ‘gfg’ is coached by a teacher. Teacher teaches her all possible topics which might appear for exam. Hence, she tends to commit very less mistakes in exam, thus performing well.

If more training data are used to construct a classifier, it qualifies any data used from test set, to test it (classifier).

If more number of data items are present in test set, such that they are used to test classifier built using training set. We can observe more accurate evaluation of classifier with respect to it’s accuracy, error rate and estimation.

**Example –**  
A student ‘gfg’ is coached by a teacher. Teacher teaches her some topics, which might appear for the exam. If the student ‘gfg’ is given a number of exams on basis of this coaching, an accurate determination of student’s weak and strong points can be found out.

If more test data are used to evaluate constructed classifier, it’s error rate, error estimate and accuracy can be accurately determined.

**Problem :**  
During partitioning of whole data set into 2 parts i.e., training set and test set, if all data items belonging to class – GFG1, are placed in test set entirely, such that none of data items of class GFG1 are in training set. It is evident, that model/classifier built, is not trained using data items of class – GFG1.

**Solution :**  
Stratification is a technique, using which data items belonging to class – GFG1 are divided and placed into two data sets i.e training set and test set, equally. Such that, model/classifier is trained by data items belonging to class -GFG1.

**Example –**  
All the four data items belonging to class – GFG1, here, are divided equally and placed, two data items each, into two data sets – training set and test set

2. Cross-validation by tenfold

# Cross Validation in Machine Learning

In **machine learning**, we couldn’t fit the model on the training data and can’t say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. For this purpose, we use the **cross-validation technique**. In this article, we’ll delve into the process of cross-validation in machine learning.

## What is Cross-Validation?

Cross validation is a technique used in machine learning to evaluate the performance of a model on unseen data. It involves dividing the available data into multiple folds or subsets, using one of these folds as a validation set, and training the model on the remaining folds. This process is repeated multiple times, each time using a different fold as the validation set. Finally, the results from each validation step are averaged to produce a more robust estimate of the model’s performance. Cross validation is an important step in the [machine learning](https://www.geeksforgeeks.org/machine-learning/) process and helps to ensure that the model selected for deployment is robust and generalizes well to new data.

## What is cross-validation used for?

The main purpose of cross validation is to prevent [overfitting](https://www.geeksforgeeks.org/overfitting-and-regularization-in-ml/), which occurs when a model is trained too well on the training data and performs poorly on new, unseen data. By evaluating the model on multiple validation sets, cross validation provides a more realistic estimate of the model’s generalization performance, i.e., its ability to perform well on new, unseen data.

## Types of Cross-Validation

There are several types of cross validation techniques, including **k-fold cross validation, leave-one-out cross validation, and Holdout validation, Stratified Cross-Validation.**The choice of technique depends on the size and nature of the data, as well as the specific requirements of the modeling problem.

### ****1. Holdout Validation****

In[Holdout Validation](https://www.geeksforgeeks.org/introduction-of-holdout-method/), we perform training on the 50% of the given dataset and rest 50% is used for the testing purpose. It’s a simple and quick way to evaluate a model. The major drawback of this method is that we perform training on the 50% of the dataset, it may possible that the remaining 50% of the data contains some important information which we are leaving while training our model i.e. higher bias.

### ****2. LOOCV (Leave One Out Cross Validation)****

In this method, we perform training on the whole dataset but leaves only one data-point of the available dataset and then iterates for each data-point. In [LOOCV](https://www.geeksforgeeks.org/loocvleave-one-out-cross-validation-in-r-programming/), the model is trained on  samples and tested on the one omitted sample, repeating this process for each data point in the dataset. It has some advantages as well as disadvantages also.

**An advantage** of using this method is that we make use of all data points and hence it is low bias.

The major**drawback**of this method is that it leads to **higher variation**in the testing model as we are testing against one data point. If the data point is an outlier it can lead to higher variation. Another drawback is it **takes a lot of execution time** as it iterates over ‘the number of data points’ times.

### ****3. Stratified Cross-Validation****

It is a technique used in machine learning to ensure that each fold of the cross-validation process maintains the same class distribution as the entire dataset. This is particularly important when dealing with imbalanced datasets, where certain classes may be underrepresented. In this method,

1. The dataset is divided into k folds while maintaining the proportion of classes in each fold.
2. During each iteration, one-fold is used for testing, and the remaining folds are used for training.
3. The process is repeated k times, with each fold serving as the test set exactly once.

[Stratified Cross-Validation](https://www.geeksforgeeks.org/stratified-k-fold-cross-validation/)is essential when dealing with classification problems where maintaining the balance of class distribution is crucial for the model to generalize well to unseen data.

### ****4. K-Fold Cross Validation****

In [K-Fold Cross Validation](https://www.geeksforgeeks.org/k-fold-cross-validation-in-r-programming/), we split the dataset into k number of subsets (known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.

***Note:****It is always suggested that the value of k should be 10 as the lower value of k is takes towards validation and higher value of k leads to LOOCV method.*

#### ****Example**** of K Fold Cross Validation

The diagram below shows an example of the training subsets and evaluation subsets generated in k-fold cross-validation. Here, we have total 25 instances. In first iteration we use the first 20 percent of data for evaluation, and the remaining 80 percent for training ([1-5] testing and [5-25] training) while in the second iteration we use the second subset of 20 percent for evaluation, and the remaining three subsets of the data for training ([5-10] testing and [1-5 and 10-25] training), and so on.

Total instances: 25  
Value of k : 5   
No. Iteration Training set observations Testing set observations  
 1 [ 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24] [0 1 2 3 4]  
 2 [ 0 1 2 3 4 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24] [5 6 7 8 9]  
 3 [ 0 1 2 3 4 5 6 7 8 9 15 16 17 18 19 20 21 22 23 24] [10 11 12 13 14]  
 4 [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 20 21 22 23 24] [15 16 17 18 19]  
 5 [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19] [20 21 22 23 24]

## Comparison between cross-validation and hold out method

### ****Advantages of train/test split:****

1. This runs K times faster than Leave One Out cross-validation because K-fold cross-validation repeats the train/test split K-times.
2. Simpler to examine the detailed results of the testing process.

**Advantages of cross-validation:**

1. More accurate estimate of out-of-sample accuracy.
2. More “efficient” use of data as every observation is used for both training and testing.

## Advantages and Disadvantages of Cross Validation

### Advantages:

1. Overcoming Overfitting: Cross validation helps to prevent overfitting by providing a more robust estimate of the model’s performance on unseen data.
2. Model Selection: Cross validation can be used to compare different models and select the one that performs the best on average.
3. Hyperparameter tuning: Cross validation can be used to optimize the hyperparameters of a model, such as the regularization parameter, by selecting the values that result in the best performance on the validation set.
4. Data Efficient: Cross validation allows the use of all the available data for both training and validation, making it a more data-efficient method compared to traditional validation techniques.

### Disadvantages:

1. Computationally Expensive: Cross validation can be computationally expensive, especially when the number of folds is large or when the model is complex and requires a long time to train.
2. Time-Consuming: Cross validation can be time-consuming, especially when there are many hyperparameters to tune or when multiple models need to be compared.
3. Bias-Variance Tradeoff: The choice of the number of folds in cross validation can impact the bias-variance tradeoff, i.e., too few folds may result in high variance, while too many folds may result in high bias.
4. Adjusting the parameters

# Hyperparameter tuning

A Machine Learning model is defined as a mathematical model with several parameters that need to be learned from the data. By training a model with existing data, we can fit the model parameters.   
However, there is another kind of parameter, known as ***Hyperparameters***, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn. This article aims to explore various strategies to tune hyperparameters for Machine learning models.

## Hyperparameter Tuning

Hyperparameter tuning is the process of selecting the optimal values for a [machine learning](https://www.geeksforgeeks.org/machine-learning/) model’s hyperparameters. Hyperparameters are settings that control the learning process of the model, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

### What are Hyperparameters?

In the context of machine learning, hyperparameters are configuration variables that are set before the training process of a model begins. They control the learning process itself, rather than being learned from the data. Hyperparameters are often used to tune the performance of a model, and they can have a significant impact on the model’s accuracy, generalization, and other metrics.

### Different Ways of Hyperparameters Tuning

Hyperparameters are configuration variables that control the learning process of a machine learning model. They are distinct from model parameters, which are the weights and biases that are learned from the data. There are several different types of hyperparameters:

#### Hyperparameters in Neural Networks

[Neural networks](https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/) have several essential hyperparameters that need to be adjusted, including:

* **Learning rate:** This hyperparameter controls the step size taken by the optimizer during each iteration of training. Too small a learning rate can result in slow convergence, while too large a learning rate can lead to instability and divergence.
* **Epochs:** This hyperparameter represents the number of times the entire training dataset is passed through the model during training. Increasing the number of epochs can improve the model’s performance but may lead to overfitting if not done carefully.
* **Number of layers:** This hyperparameter determines the depth of the model, which can have a significant impact on its complexity and learning ability.
* **Number of nodes per layer:** This hyperparameter determines the width of the model, influencing its capacity to represent complex relationships in the data.
* **Architecture:** This hyperparameter determines the overall structure of the neural network, including the number of layers, the number of neurons per layer, and the connections between layers. The optimal architecture depends on the complexity of the task and the size of the dataset
* **Activation function:** This hyperparameter introduces non-linearity into the model, allowing it to learn complex decision boundaries. Common activation functions include sigmoid, tanh, and Rectified Linear Unit (ReLU).

#### Hyperparameters in Support Vector Machine

We take into account some essential hyperparameters for fine-tuning [SVMs](https://www.geeksforgeeks.org/support-vector-machine-algorithm/):

* **C:** The regularization parameter that controls the trade-off between the margin and the number of training errors. A larger value of C penalizes training errors more heavily, resulting in a smaller margin but potentially better generalization performance. A smaller value of C allows for more training errors but may lead to overfitting.
* **Kernel:** The kernel function that defines the similarity between data points. Different kernels can capture different relationships between data points, and the choice of kernel can significantly impact the performance of the SVM. Common kernels include linear, polynomial, radial basis function (RBF), and sigmoid.
* **Gamma:** The parameter that controls the influence of support vectors on the decision boundary. A larger value of gamma indicates that nearby support vectors have a stronger influence, while a smaller value indicates that distant support vectors have a weaker influence. The choice of gamma is particularly important for RBF kernels.

#### Hyperparameters in XGBoost

The following essential [XGBoost](https://www.geeksforgeeks.org/xgboost/" \t "_blank) hyperparameters need to be adjusted:

* **learning\_rate:** This hyperparameter determines the step size taken by the optimizer during each iteration of training. A larger learning rate can lead to faster convergence, but it may also increase the risk of overfitting. A smaller learning rate may result in slower convergence but can help prevent overfitting.
* **n\_estimators:** This hyperparameter determines the number of boosting trees to be trained. A larger number of trees can improve the model’s accuracy, but it can also increase the risk of overfitting. A smaller number of trees may result in lower accuracy but can help prevent overfitting.
* **max\_depth:** This hyperparameter determines the maximum depth of each tree in the ensemble. A larger max\_depth can allow the trees to capture more complex relationships in the data, but it can also increase the risk of overfitting. A smaller max\_depth may result in less complex trees but can help prevent overfitting.
* **min\_child\_weight:** This hyperparameter determines the minimum sum of instance weight (hessian) needed in a child node. A larger min\_child\_weight can help prevent overfitting by requiring more data to influence the splitting of trees. A smaller min\_child\_weight may allow for more aggressive tree splitting but can increase the risk of overfitting.
* **subsample:** This hyperparameter determines the percentage of rows used for each tree construction. A smaller subsample can improve the efficiency of training but may reduce the model’s accuracy. A larger subsample can increase the accuracy but may make training more computationally expensive.

Some other examples of model hyperparameters include:

1. The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization
2. Number of Trees and Depth of Trees for Random Forests.
3. The learning rate for training a neural network.
4. Number of Clusters for Clustering Algorithms.
5. The k in k-nearest neighbors.

### Hyperparameter Tuning techniques

Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. The two best strategies for Hyperparameter tuning are:

1. [GridSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)
2. [RandomizedSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html)
3. [Bayesian Optimization](https://www.geeksforgeeks.org/catboost-bayesian-optimization/)

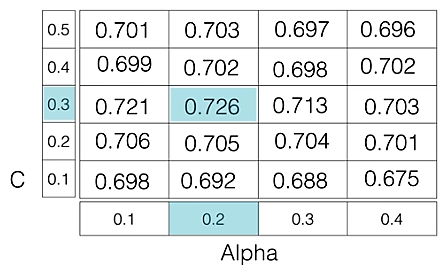
### ****1. GridSearchCV****

Grid search can be considered as a “brute force” approach to hyperparameter optimization. We fit the model using all possible combinations after creating a grid of potential discrete hyperparameter values. We log each set’s model performance and then choose the combination that produces the best results. This approach is called GridSearchCV, because it searches for the best set of hyperparameters from a grid of hyperparameters values.

An exhaustive approach that can identify the ideal hyperparameter combination is grid search. But the slowness is a disadvantage. It often takes a lot of processing power and time to fit the model with every potential combination, which might not be available.

**For example:**if we want to set two hyperparameters C and Alpha of the Logistic Regression Classifier model, with different sets of values. The grid search technique will construct many versions of the model with all possible combinations of hyperparameters and will return the best one.

As in the image, for C = [0.1, 0.2, 0.3, 0.4, 0.5] and Alpha = [0.1, 0.2, 0.3, 0.4]. For a combination of ***C=0.3 and Alpha=0.2***, the performance score comes out to be **0.726(Highest)**, therefore it is selected.



The following code illustrates how to use GridSearchCV

11. Define the following terms:

1. Purity vs. Silhouette width

# Silhouette Coefficient

# After learning and applying several supervised ML algorithms like least square regression, logistic regression, SVM, decision tree etc. most of us try to have some hands-on unsupervised learning by implementing some clustering techniques like K-Means, DBSCAN or HDBSCAN.

We usually start with K-Means clustering. After going through several tutorials and Medium stories you will be able to implement k-means clustering easily. But as you implement it, a question starts to bug your mind: how can we measure its goodness of fit? Supervised algorithms have lots of metrics to check their goodness of fit like accuracy, r-square value, sensitivity, specificity etc. but what can we calculate to measure the accuracy or goodness of our clustering technique? The answer to this question is Silhouette Coefficient or Silhouette score.

# ****Silhouette Coefficient:****

Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.

1: Means clusters are well apart from each other and clearly distinguished.

0: Means clusters are indifferent, or we can say that the distance between clusters is not significant.

-1: Means clusters are assigned in the wrong way.

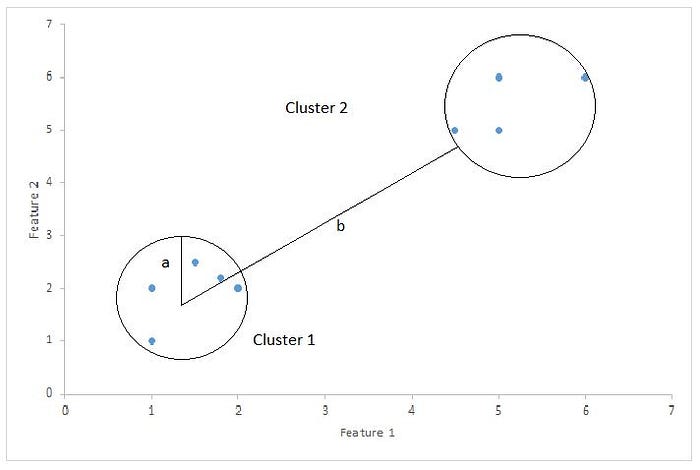


Image by author

Silhouette Score = (b-a)/max(a,b)

where

a= average intra-cluster distance i.e the average distance between each point within a cluster.

b= average inter-cluster distance i.e the average distance between all clusters.

# Calculating Silhouette Score

**Importing libraries:**

import pandas as pd  
import numpy as np  
import seaborn as sns  
from sklearn.cluster import KMeans  
from sklearn.metrics import silhouette\_score  
%matplotlib inline

**Generating some random data:**

To run clustering algorithm we are generating 100 random points.

X= np.random.rand(50,2)  
Y= 2 + np.random.rand(50,2)  
Z= np.concatenate((X,Y))  
Z=pd.DataFrame(Z) #converting into data frame for ease

Plotting the data:

sns.scatterplot(Z[0],Z[1])

Output

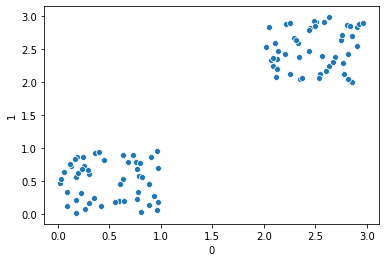


Image by author

**Applying KMeans Clustering with 2 clusters:**

KMean= KMeans(n\_clusters=2)  
KMean.fit(Z)  
label=KMean.predict(Z)

**Calculating the silhouette score:**

print(f'Silhouette Score(n=2): {silhouette\_score(Z, label)}')

Output: Silhouette Score(n=2): 0.8062146115881652

We can say that the clusters are well apart from each other as the silhouette score is closer to 1.

To check whether our silhouette score is providing the right information or not let’s create another scatter plot showing labelled data points.

sns.scatterplot(Z[0],Z[1],hue=label)

Output:

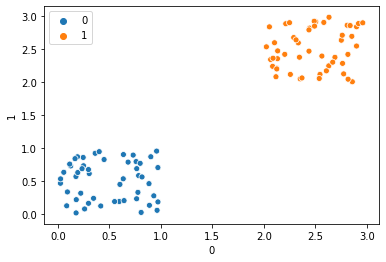


Image by author

It can be seen clearly in the above figure that each cluster is well apart from each other.

Let’s try with 3 clusters:

KMean= KMeans(n\_clusters=3)  
KMean.fit(Z)  
label=KMean.predict(Z)  
print(f’Silhouette Score(n=3): {silhouette\_score(Z, label)}’)  
sns.scatterplot(Z[0],Z[1],hue=label,palette=’inferno\_r’)

Output:

Silhouette Score(n=3): 0.5969732708311737

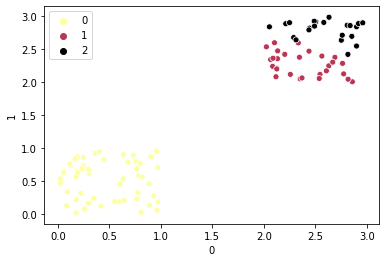


Image by author

As you can see in the above figure clusters are not well apart. The inter cluster distance between cluster 1 and cluster 2 is almost negligible. That is why the silhouette score for n= 3(0.596) is lesser than that of n=2(0.806).

When dealing with higher dimensions, the silhouette score is quite useful to validate the working of clustering algorithm as we can’t use any type of visualization to validate clustering when dimensions are greater than 3.

We can also use the silhouette score to check the optimal number of clusters. In the above example, we can say that the optimal number of clusters is 2 as its silhouette score is greater than that of 3 clusters.

1. Boosting vs. Bagging

# Bagging vs Boosting in Machine Learning

As we know, [Ensemble learning](https://www.geeksforgeeks.org/ensemble-classifier-data-mining/) helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote. **Bagging** and **Boosting**are two types of **Ensemble Learning**. These two decrease the variance of a single estimate as they combine several estimates from different models. So the result may be a model with higher stability. Let’s understand these two terms in a glimpse.

1. **Bagging**: It is a homogeneous weak learners’ model that learns from each other independently in parallel and combines them for determining the model average.
2. **Boosting**: It is also a homogeneous weak learners’ model but works differently from Bagging. In this model, learners learn sequentially and adaptively to improve model predictions of a learning algorithm.

Let’s look at both of them in detail and understand the Difference between Bagging and Boosting.

## Bagging

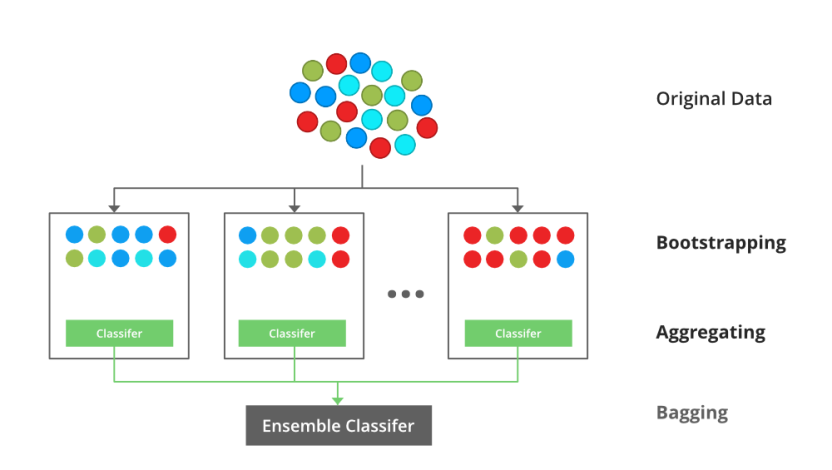
**B**ootstrap **A**ggregating, also known as bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It decreases the [variance](https://www.geeksforgeeks.org/mathematics-mean-variance-and-standard-deviation/)and helps to avoid [overfitting](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/). It is usually applied to [decision tree methods](https://www.geeksforgeeks.org/decision-tree/). Bagging is a special case of the model averaging approach.

**Description of the Technique**

Suppose a set D of d tuples, at each iteration i, a training set Di of d tuples is selected via row sampling with a replacement method (i.e., there can be repetitive elements from different d tuples) from D (i.e., bootstrap). Then a classifier model Mi is learned for each training set D < i. Each classifier Mi returns its class prediction. The bagged classifier M\* counts the votes and assigns the class with the most votes to X (unknown sample).

**Implementation Steps of Bagging**

* **Step 1:** Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
* **Step 2:** A base model is created on each of these subsets.
* **Step 3:**Each model is learned in parallel with each training set and independent of each other.
* **Step 4:**The final predictions are determined by combining the predictions from all the models.



*An illustration for the concept of bootstrap aggregating (Bagging)*

**Example of Bagging**

The [Random Forest model](https://www.geeksforgeeks.org/random-forest-regression-in-python/) uses Bagging, where decision tree models with higher variance are present. It makes random feature selection to grow trees. Several random trees make a Random Forest.

*To read more refer to this article:* [Bagging classifier](https://www.geeksforgeeks.org/ml-bagging-classifier/)

## Boosting

Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models is added.

**Boosting Algorithms**

There are several boosting algorithms. The original ones, proposed by **Robert Schapire** and **Yoav Freund** were not adaptive and could not take full advantage of the weak learners. Schapire and Freund then developed [AdaBoost](https://www.geeksforgeeks.org/implementing-the-adaboost-algorithm-from-scratch/), an adaptive boosting algorithm that won the prestigious Gödel Prize. AdaBoost was the first really successful boosting algorithm developed for the purpose of binary classification. AdaBoost is short for Adaptive Boosting and is a very popular boosting technique that combines multiple “weak classifiers” into a single “strong classifier”.

***Algorithm:***

1. *Initialise the dataset and assign equal weight to each of the data point.*
2. *Provide this as input to the model and identify the wrongly classified data points.*
3. *Increase the weight of the wrongly classified data points and decrease the weights of correctly classified data points. And then normalize the weights of all data points.*
4. *if (got required results)  
     Goto step 5  
   else  
     Goto step 2*
5. *End*



*To read more refer to this article:*[Boosting and AdaBoost in ML](https://www.geeksforgeeks.org/boosting-in-machine-learning-boosting-and-adaboost/)

### ****Similarities Between Bagging and Boosting****

Bagging and Boosting, both being the commonly used methods, have a universal similarity of being classified as ensemble methods. Here we will explain the similarities between them.

1. Both are ensemble methods to get N learners from 1 learner.
2. Both generate several training data sets by random sampling.
3. Both make the final decision by averaging the N learners (or taking the majority of them i.e Majority Voting).
4. Both are good at reducing variance and provide higher stability.

### ****Differences Between Bagging and Boosting****

| **S.NO** | **Bagging** | **Boosting** |
| --- | --- | --- |
| **1.** | **The simplest way of combining predictions that  belong to the same type.** | **A way of combining predictions that  belong to the different types.** |
| **2.** | **Aim to decrease variance, not bias.** | **Aim to decrease bias, not variance.** |
| **3.** | **Each model receives equal weight.** | **Models are weighted according to their performance.** |
| **4.** | **Each model is built independently.** | **New models are influenced  by the performance of previously built models.** |
| **5.** | **Different training data subsets are selected using row sampling with replacement and random sampling methods from the entire training dataset.** | **Every new subset contains the elements that were misclassified by previous models.** |
| **6.** | **Bagging tries to solve the over-fitting problem.** | **Boosting tries to reduce bias.** |
| **7.** | **If the classifier is unstable (high variance), then apply bagging.** | **If the classifier is stable and simple (high bias) the apply boosting.** |
| **8.** | **In this base classifiers are trained parallelly.** | **In this base classifiers are trained sequentially.** |
| **9** | **Example: The Random forest model uses Bagging.** | **Example: The AdaBoost uses Boosting techniques** |

1. The eager learner vs. the lazy learner

## Lazy learning defers the computation of predictions until needed, relying on instance-specific information, while eager learning precomputes a model during training, making predictions faster but potentially requiring more memory.

Lazy learning and eager learning are two contrasting approaches in machine learning, primarily referring to the handling of model construction and prediction. Let’s delve into the details of the differences between lazy and eager learning:

| **Aspect** | **Lazy Learning** | **Eager Learning** |
| --- | --- | --- |
| **Timing of Model Building** | The model is built during prediction. | The model is built before prediction. |
| **Data Dependency** | Relies heavily on the training data during prediction. | Less dependent on training data during prediction. |
| **Computational Efficiency** | Faster during training, but slower during prediction due to real-time model building. | Slower during training, but faster during prediction due to pre-built model. |
| **Example** | k-Nearest Neighbors (KNN) | Decision Trees, Support Vector Machines (SVM), Neural Networks |
| **Memory Usage** | Less memory usage during training, but more during prediction. | More memory usage during training, but less during prediction. |