Universidad Politécnica de Yucatán.

Student: Julio Emanuel González Gonzalez

Degree: Robotics

Grade and group: 9°A

Subject: Machine Learning

Professor: Victor Ortiz

Activity: Solution to most common problems in ML

Overfitting & Underfitting

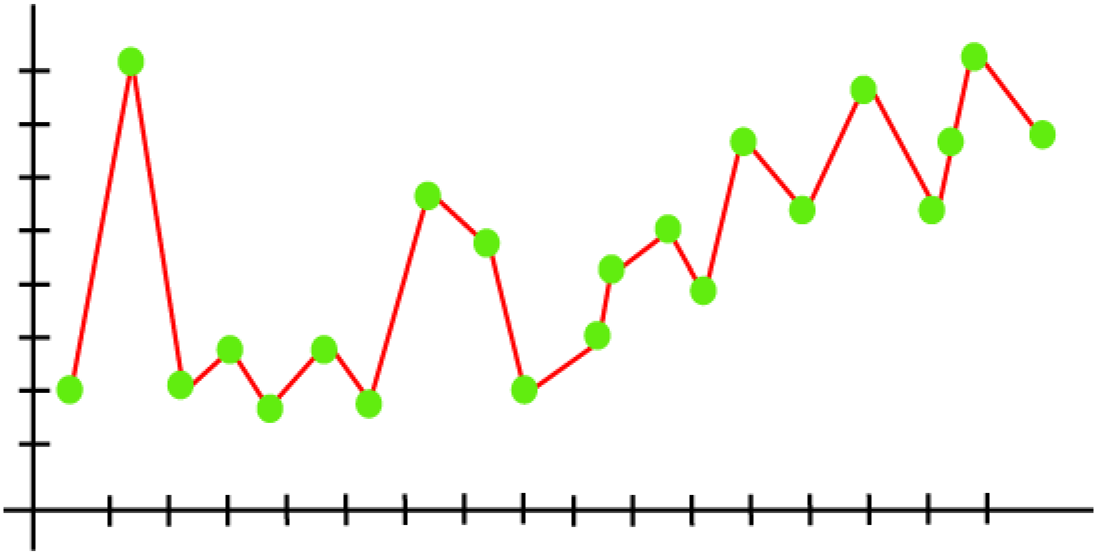
Overfitting

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

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

Overfitting is the main problem that occurs in supervised learning.

Example: The concept of the overfitting can be understood by the below graph of the linear regression output:



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

How to avoid the Overfitting in Model

Both overfitting and underfitting cause the degraded performance of the machine learning model. But the main cause is overfitting, so there are some ways by which we can reduce the occurrence of overfitting in our model.

- Cross-Validation
- Training with more data
- Removing features
- Early stopping the training
- Regularization
- Assembling

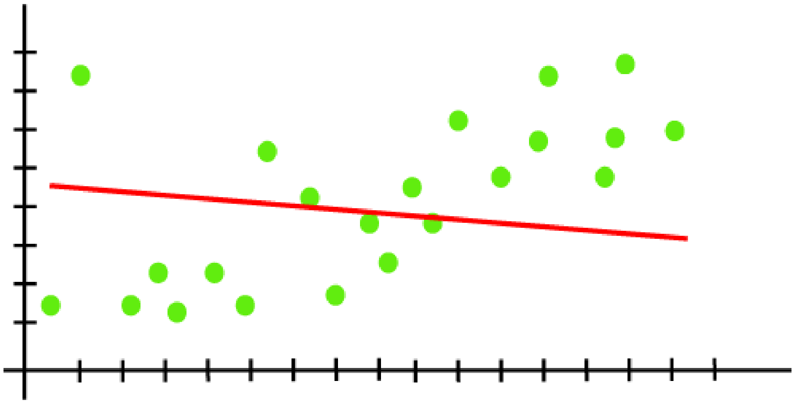
Underfitting

Underfitting occurs when our machine learning model is not able to capture the underlying trend of the data. To avoid the overfitting in the model, the fed of training data can be stopped at an early stage, due to which the model may not learn enough from the training data. As a result, it may fail to find the best fit of the dominant trend in the data.

In the case of underfitting, the model is not able to learn enough from the training data, and hence it reduces the accuracy and produces unreliable predictions.

An underfitted model has high bias and low variance.

Example: We can understand the underfitting using below output of the linear regression model:



As we can see from the above diagram, the model is unable to capture the data points present in the plot.

How to avoid underfitting:

- By increasing the training time of the model.
- By increasing the number of features.

Goodness of Fit

The "Goodness of fit" term is taken from the statistics, and the goal of the machine learning models to achieve the goodness of fit. In statistics modeling, it defines how closely the result or predicted values match the true values of the dataset.

The model with a good fit is between the underfitted and overfitted model, and ideally, it makes predictions with 0 errors, but in practice, it is difficult to achieve it.

As when we train our model for a time, the errors in the training data go down, and the same happens with test data. But if we train the model for a long duration, then the performance of the model may decrease due to the overfitting, as the model also learn the noise present in the dataset. The errors in the test dataset start increasing, so the point, just before the raising of errors, is the good point, and we can stop here for achieving a good model.

There are two other methods by which we can get a good point for our model, which are the resampling method to estimate model accuracy and validation dataset.

Outliers

Definition of outliers

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Before abnormal observations can be singled out, it is necessary to characterize normal observations.

Ways to describe data.

Two activities are essential for characterizing a set of data:

- Examination of the overall shape of the graphed data for important features, including symmetry and departures from assumptions. The chapter on Exploratory Data Analysis (EDA) discusses assumptions and summarization of data in detail.
- Examination of the data for unusual observations that are far removed from the mass of data. These points are often referred to as outliers. Two graphical techniques for identifying outliers, scatter plots and box plots, along with an analytic procedure for detecting outliers when the distribution is normal (Grubbs' Test), are also discussed in detail in the EDA chapter.

Box plot construction.

The box plot is a useful graphical display for describing the behavior of the data in the middle as well as at the ends of the distributions. The box plot uses the median and the lower and upper quartiles (defined as the 25th and 75th percentiles). If the lower quartile is Q1 and the upper quartile is Q3, then the difference (Q3 - Q1) is called the interquartile range or IQ.

Box plots with fences.

A box plot is constructed by drawing a box between the upper and lower quartiles with a solid line drawn across the box to locate the median. The following quantities (called fences) are needed for identifying extreme values in the tails of the distribution:

- lower inner fence: Q1 - 1.5*IQ
- upper inner fence: Q3 + 1.5*IQ
- lower outer fence: Q1 - 3*IQ
- upper outer fence: Q3 + 3*IQ

Outlier detection criteria.

A point beyond an inner fence on either side is considered a mild outlier. A point beyond an outer fence is considered an extreme outlier.

Example of an outlier box plot.

The data set of N = 90 ordered observations as shown below is examined for outliers:

30, 171, 184, 201, 212, 250, 265, 270, 272, 289, 305, 306, 322, 322, 336, 346, 351, 370, 390, 404, 409, 411, 436, 437, 439, 441, 444, 448, 451, 453, 470, 480, 482, 487, 494, 495, 499, 503, 514, 521, 522, 527, 548, 550, 559, 560, 570, 572, 574, 578, 585, 592, 592, 607, 616, 618, 621, 629, 637, 638, 640, 656, 668, 707, 709, 719, 737, 739, 752, 758, 766, 792, 792, 794, 802, 818, 830, 832, 843, 858, 860, 869, 918, 925, 953, 991, 1000, 1005, 1068, 1441

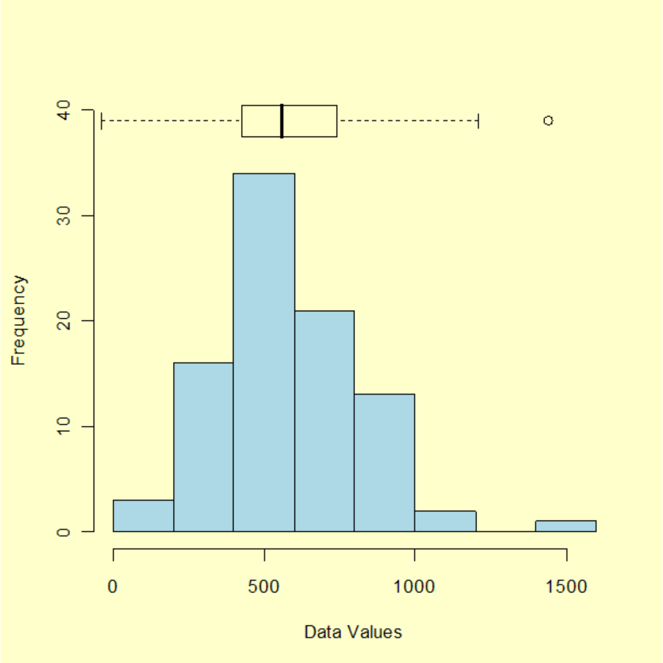
The above data is available as a text file.

The computations are as follows:

- Median = (n+1)/2 largest data point = the average of the 45th and 46th ordered points = (559 + 560)/2 = 559.5
- Lower quartile = .25(N+1)th ordered point = 22.75th ordered point = 411 + .75(436-411) = 429.75
- Upper quartile = .75(N+1)th ordered point = 68.25th ordered point = 739 +.25(752-739) = 742.25
- Interquartile range = 742.25 - 429.75 = 312.5
- Lower inner fence = 429.75 - 1.5 (312.5) = -39.0
- Upper inner fence = 742.25 + 1.5 (312.5) = 1211.0
- Lower outer fence = 429.75 - 3.0 (312.5) = -507.75
- Upper outer fence = 742.25 + 3.0 (312.5) = 1679.75

From an examination of the fence points and the data, one point (1441) exceeds the upper inner fence and stands out as a mild outlier; there are no extreme outliers.

A histogram with an overlaid box plot is shown below.



The outlier is identified as the largest value in the data set, 1441, and appears as the circle to the right of the box plot.

Outliers should be investigated carefully. Often, they contain valuable information about the process under investigation or the data gathering and recording process. Before considering the possible elimination of these points from the data, one should try to understand why they appeared and whether it is likely similar values will continue to appear. Of course, outliers are often bad data points.

Dimensionality Reduction problem.

Dimensionality reduction is a technique used to reduce the number of features in a dataset while retaining as much of the important information as possible. In other words, it is a process of transforming high-dimensional data into a lower-dimensional space that still preserves the essence of the original data.

In machine learning, high-dimensional data refers to data with a large number of features or variables. The curse of dimensionality is a common problem in machine learning, where the performance of the model deteriorates as the number of features increases. This is because the complexity of the model increases with the number of features, and it becomes more difficult to find a good solution. In addition, high-dimensional data can also lead to overfitting, where the model fits the training data too closely and does not generalize well to new data.

Dimensionality reduction can help to mitigate these problems by reducing the complexity of the model and improving its generalization performance. There are two main approaches to dimensionality reduction: feature selection and feature extraction.

Feature Selection:

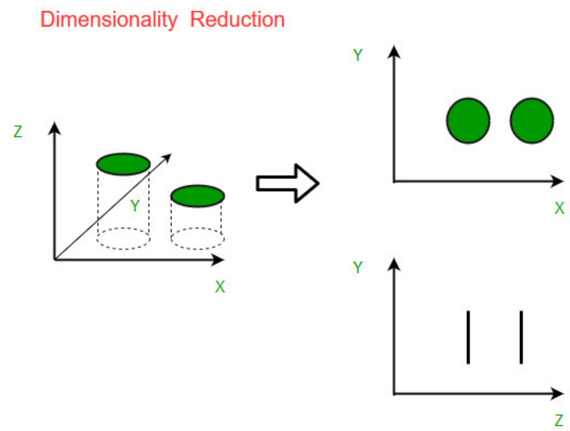
Feature selection involves selecting a subset of the original features that are most relevant to the problem at hand. The goal is to reduce the dimensionality of the dataset while retaining the most important features. There are several methods for feature selection, including filter methods, wrapper methods, and embedded methods. Filter methods rank the features based on their relevance to the target variable, wrapper methods use the model performance as the criteria for selecting features, and embedded methods combine feature selection with the model training process.

Feature Extraction:

Feature extraction involves creating new features by combining or transforming the original features. The goal is to create a set of features that captures the essence of the original data in a lower-dimensional space. There are several methods for feature extraction, including principal component analysis (PCA), linear discriminant analysis (LDA), and t-distributed stochastic neighbor embedding (t-SNE). PCA is a popular technique that projects the original features onto a lower-dimensional space while preserving as much of the variance as possible.

Importance of Dimensionality Reduction in Machine Learning and Predictive Modeling

An intuitive example of dimensionality reduction can be discussed through a simple e-mail classification problem, where we need to classify whether the e-mail is spam or not. This can involve a large number of features, such as whether or not the e-mail has a generic title, the content of the e-mail, whether the e-mail uses a template, etc. However, some of these features may overlap. In another condition, a classification problem that relies on both humidity and rainfall can be collapsed into just one underlying feature, since both aforementioned are correlated to a high degree. Hence, we can reduce the number of features in such problems. A 3-D classification problem can be hard to visualize, whereas a 2-D one can be mapped to a simple 2-dimensional space, and a 1-D problem to a simple line. The below figure illustrates this concept, where a 3-D feature space is split into two 2-D feature spaces, and later, if found to be correlated, the number of features can be reduced even further.



Components of Dimensionality Reduction

There are two components of dimensionality reduction:

Feature selection: In this, we try to find a subset of the original set of variables, or features, to get a smaller subset which can be used to model the problem. It usually involves three ways:

- Filter
- Wrapper
- Embedded

Feature extraction: This reduces the data in a high dimensional space to a lower dimension space, i.e., a space with lesser no. of dimensions.

Important points:

Dimensionality reduction is the process of reducing the number of features in a dataset while retaining as much information as possible.

This can be done to reduce the complexity of a model, improve the performance of a learning algorithm, or make it easier to visualize the data.

Techniques for dimensionality reduction include principal component analysis (PCA), singular value decomposition (SVD), and linear discriminant analysis (LDA).

Each technique projects the data onto a lower-dimensional space while preserving important information.

Dimensionality reduction is performed during pre-processing stage before building a model to improve the performance.

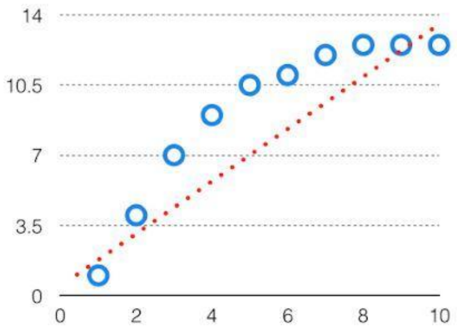
It is important to note that dimensionality reduction can also discard useful information, so care must be taken when applying these techniques.

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 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 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 of Data. This happens when the hypothesis is too simple or linear in nature. Refer to the graph given below for an example of such a situation.

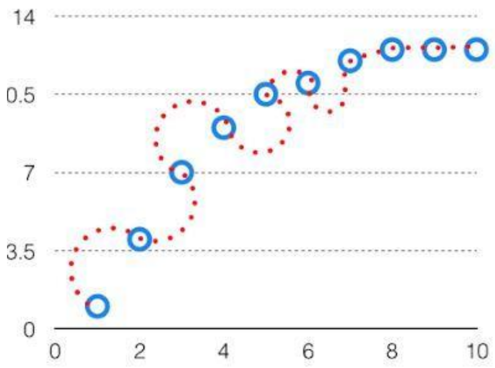


In such a problem, a hypothesis looks like follows.

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

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.

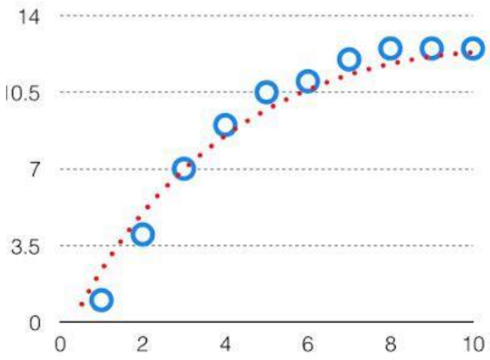


In such a problem, a hypothesis looks like follows.

$$h_\theta(x) = g\left(\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4\right)$$

Bias Variance Tradeoff

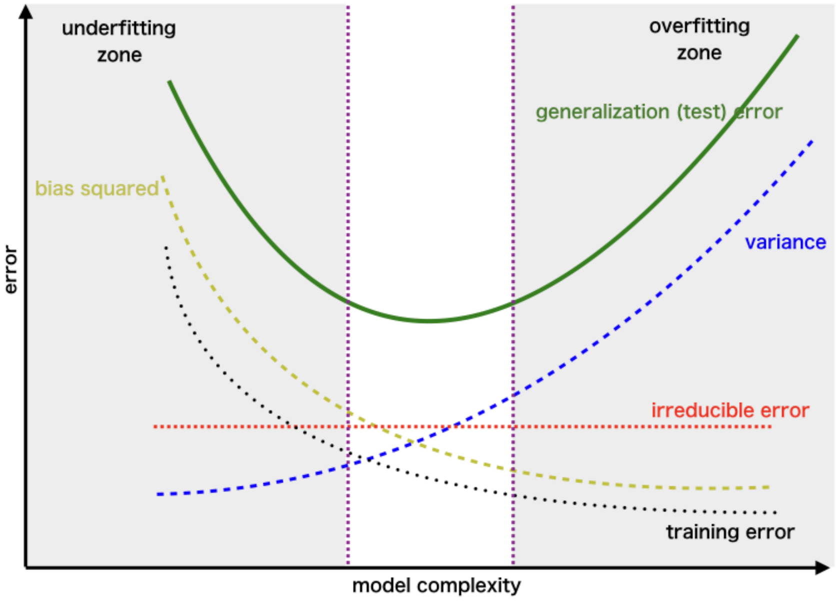
If the algorithm is too simple (hypothesis with linear equation) then it may be on high bias and low variance condition and thus is error prone. If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both 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 Tradeoff.

$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

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 –



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.

References

Bibliografía

7.1.6. What are outliers in the data? (s/f). Nist.gov. Recuperado el 16 de septiembre de 2023, de https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm

Follow, P. (2020, febrero 3). Bias-variance trade off - machine learning. GeeksforGeeks. https://www.geeksforgeeks.org/ml-bias-variance-trade-off/

Introduction to dimensionality reduction. (2017, junio 1). GeeksforGeeks. https://www.geeksforgeeks.org/dimensionality-reduction/

Overfitting and Underfitting in Machine Learning. (s/f). Www.javatpoint.com. Recuperado el 16 de septiembre de 2023, de https://www.javatpoint.com/overfitting-and-underfitting-in-machine-learning