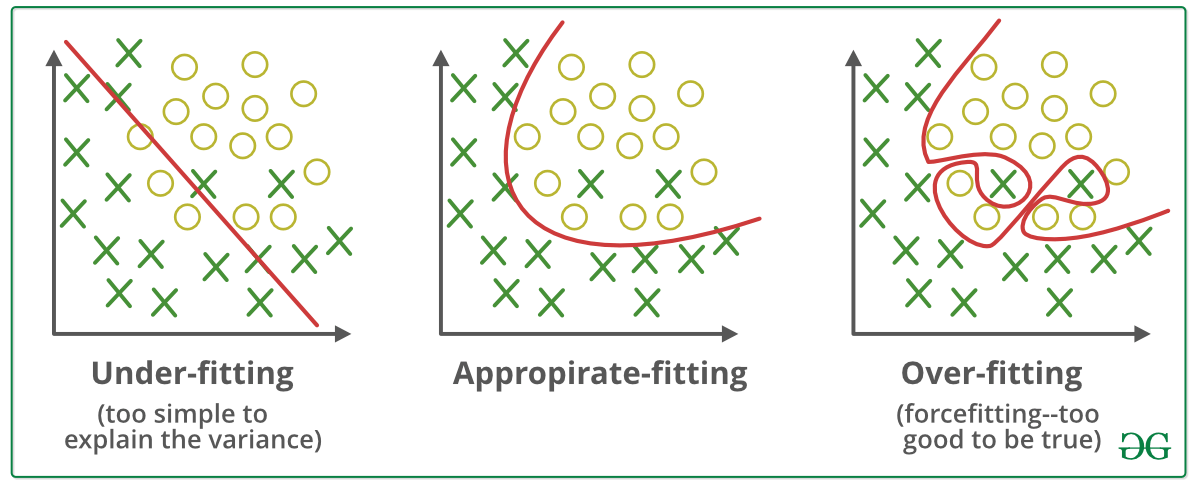
**Trình bày hiện tượng Overfitting trong Học máy và các giải pháp tránh overfitting trong đó tập trung vào các phương pháp regularization.**

**What is overfitting?**

Overfitting is a concept in data science, which occurs when a statistical model fits exactly against its training data. When this happens, the algorithm unfortunately cannot perform accurately against unseen data, defeating its purpose. Generalization of a model to new data is ultimately what allows us to use machine learning algorithms every day to make predictions and classify data.

When machine learning algorithms are constructed, they leverage a sample dataset to train the model. However, when the model trains for too long on sample data or when the model is too complex, it can start to learn the “noise,” or irrelevant information, within the dataset. When the model memorizes the noise and fits too closely to the training set, the model becomes “overfitted,” and it is unable to generalize well to new data. If a model cannot generalize well to new data, then it will not be able to perform the classification or prediction tasks that it was intended for.

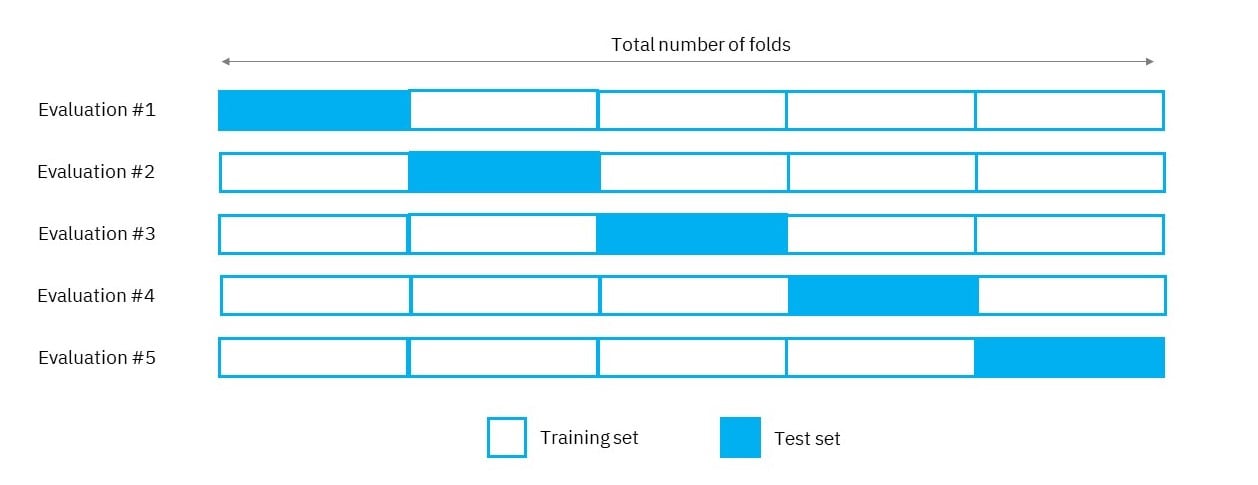
Low error rates and a high variance are good indicators of overfitting. In order to prevent this type of behavior, part of the training dataset is typically set aside as the “test set” to check for overfitting. If the training data has a low error rate and the test data has a high error rate, it signals overfitting.



## How to detect overfit models

To understand the accuracy of machine learning models, it’s important to test for model fitness. K-fold cross-validation is one of the most popular techniques to assess accuracy of the model.

In k-folds cross-validation, data is split into k equally sized subsets, which are also called “folds.” One of the k-folds will act as the test set, also known as the holdout set or validation set, and the remaining folds will train the model. This process repeats until each of the fold has acted as a holdout fold. After each evaluation, a score is retained and when all iterations have completed, the scores are averaged to assess the performance of the overall model.

For example, let’s say we divided the dataset into five sub-groups. The process can be visualized, like this: 

## How to avoid overfitting

While using a linear model helps us avoid overfitting, many real-world problems are nonlinear ones. In addition to understanding how to detect overfitting, it is important to understand how to avoid overfitting altogether. Below are a number of techniques that you can use to prevent overfitting:

* Early stopping
* Train with more data
* Data augmentation
* Feature selection
* Regulization
* Ensemble methods

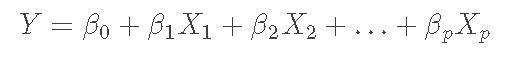
We’re going to focus on regulization method:

Regularization is a technique that adds information to a model to prevent the occurrence of overfitting. It is a type of regression that minimizes the coefficient estimates to zero to reduce the capacity (size) of a model. In this context, the reduction of the capacity of a model involves the removal of extra weights.

Regularization removes extra weights from the selected features and redistributes the weights evenly. This means that regularization discourages the learning of a model of both high complexity and flexibility. A highly flexible model is one that possesses the freedom to fit as many data points as possible.

Furthermore, in this context, we may judge the complexity of a predictive model by the number of features it possesses. A model with a lot of features to learn from is at a greater risk of overfitting. By discouraging the learning of (or use of) highly complex and flexible models, the risk of overfitting is lowered.

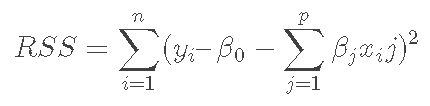
Let’s use a linear regression equation to explain regularization further.



YY represents the value that is to be predicted. βiβi stands for the regressor coefficient estimates for the corresponding predictor XiXi. And, XiXi represents the weights or magnitudes assigned to various predictors (independent variables). Here, i represents any value greater than or equal to 0, and less than p.

A loss function is involved in the fitting process. It is computed as the difference between the actual and predicted output from a model. A loss function provides a means of assessing how well an algorithm models given data. It is used to minimize the error, in turn optimizing the weights. In this context, the loss function is referred to as the residual sum of squares (RSS).

Below is the equation for the loss function.



Based on the training data, the loss function will adjust the coefficients. If the presence of noise or outliers is found in the training data, the approximated coefficients will not generalize well to the unseen data. Regularization comes into play and shrinks the learned estimates towards zero.

In other words, it tunes the loss function by adding a penalty term, that prevents excessive fluctuation of the coefficients. Thereby, reducing the chances of overfitting.