Regularization

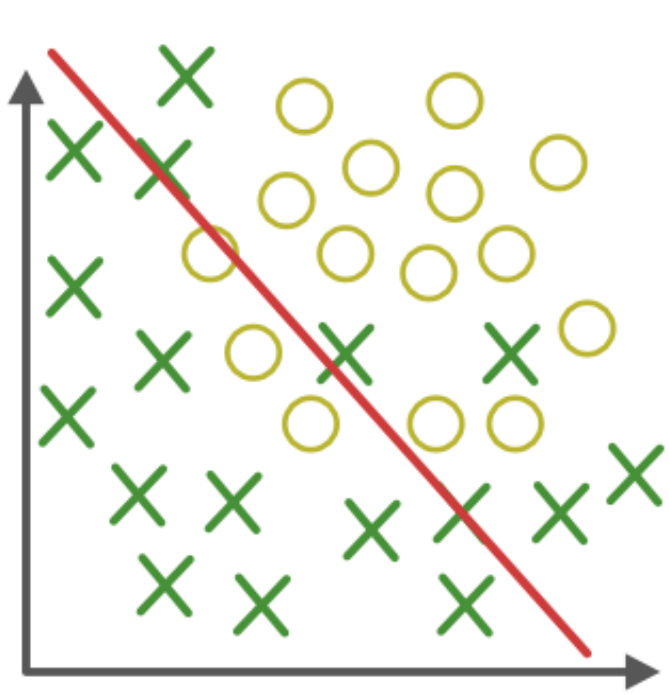
VIP Machine Learning Course

Issues with machine learning models

- Overfitting: This occurs when a machine learning model performs well on training data (low error and high accuracy) but does not generalize well to test data (high error and low accuracy).

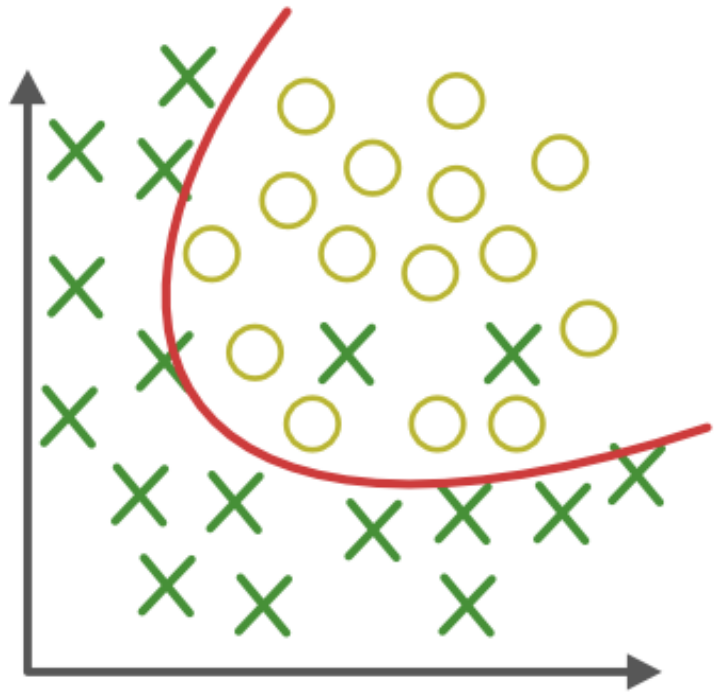
- Underfitting: This happens when a model does not learn well from training data, resulting in high error on both training and test data, and low accuracy on both sets. (The model neither learns well nor predicts well.)

Issues with machine learning models

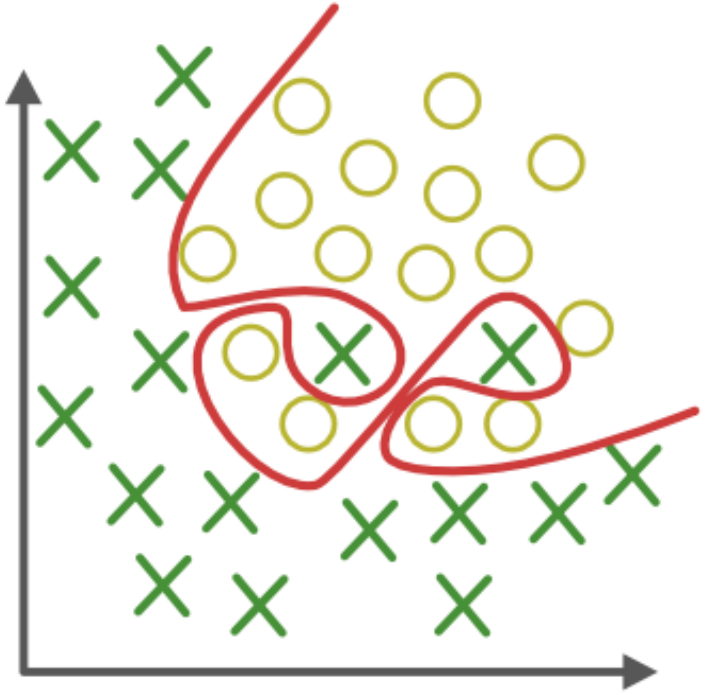


Under-fitting

(too simple to explain the variance)



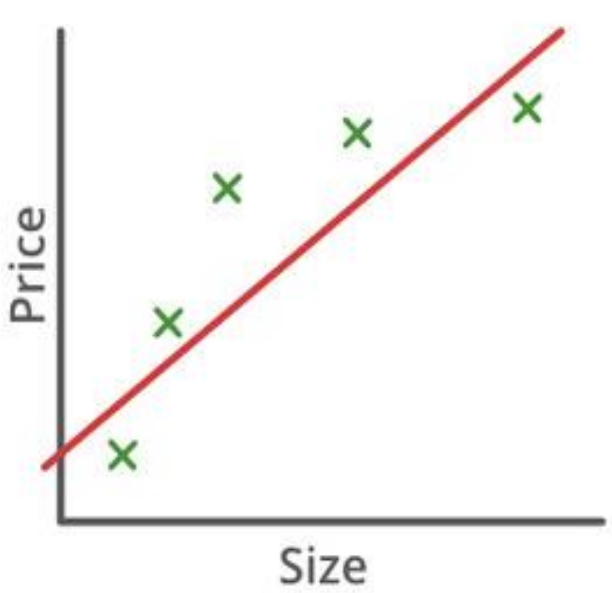
Appropirate-fitting



Over-fitting

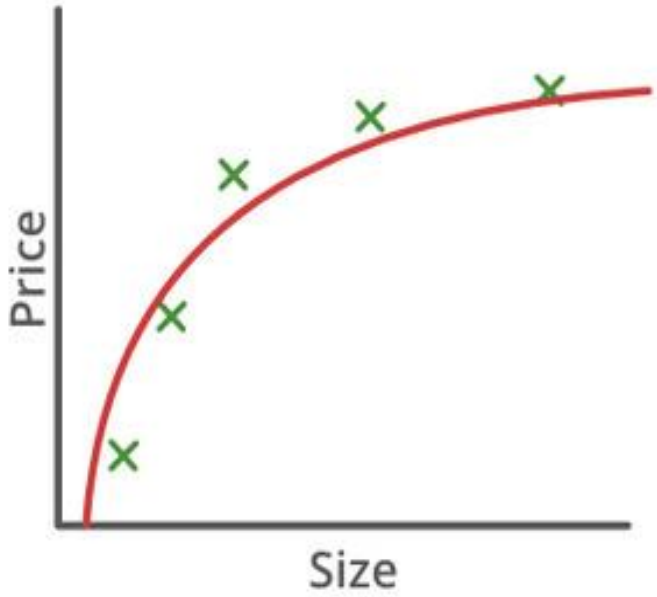
(forcefitting--too good to be true)

Issues with machine learning models



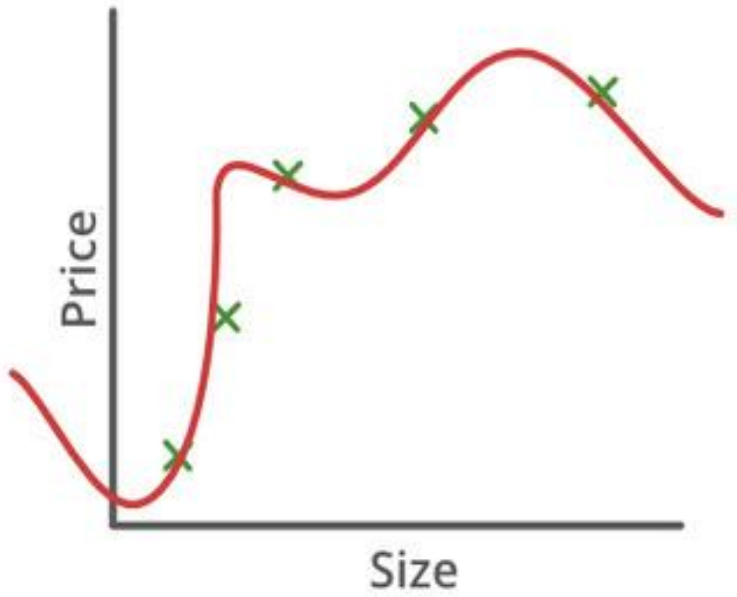
$\theta_0 + \theta_1 x$

High bais (underfit)



$\theta_0 + \theta_1 x + \theta_2 x^2$

High bais (underfit)



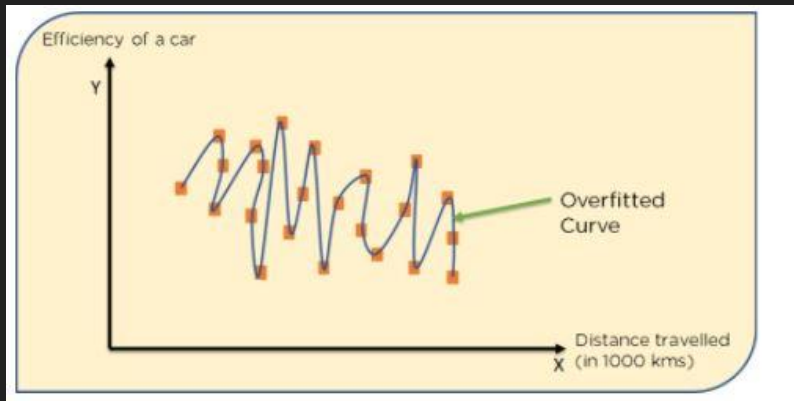
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance (overfit)

Overfitting

- Overfitting occurs when a machine learning model learns too much from the training data, leading to difficulties in generalizing well on test data and making accurate predictions. In such cases, the model also learns the noise present in the data.
- Instances when overfitting may occur include:
- The model is overly complex and incorporates collinear features, which increases data variance.
- The number of features in the data is high or equal to the number of data points (high-dimensional feature space).
- The dataset is small or very small in size.
- The data is not preprocessed properly and contains noise.

These conditions contribute to overfitting by allowing the model to fit too closely to the training data, including its noise and specific characteristics that do not generalize well to new, unseen data.



Methods to combat Overfitting

- Methods to combat Overfitting:

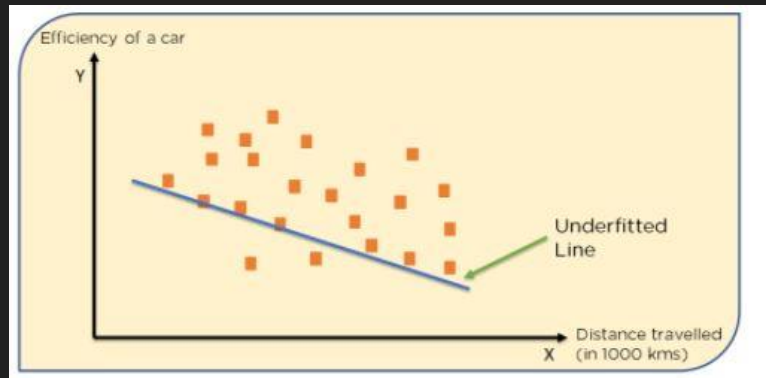
1. Applying K-fold cross validation
2. Increasing data (collecting more data and also generating synthetic data)
3. Efficient feature selection
4. Regularization (L1 and L2) and (also using Drop Out technique)

- Additional techniques for neural networks in deep learning:

1. Eliminating layers from the model (in neural networks in deep learning)
2. Early stopping during the learning process (in neural networks in deep learning)

Underfitting

- Underfitting occurs when a machine learning model is not sufficiently complex to learn meaningful relationships between features and the target variable. In this case, the model exhibits high Bias and low Variance.
- A model that suffers from Underfitting does not learn well from the training data, and consequently, it does not perform well on both training and test data.

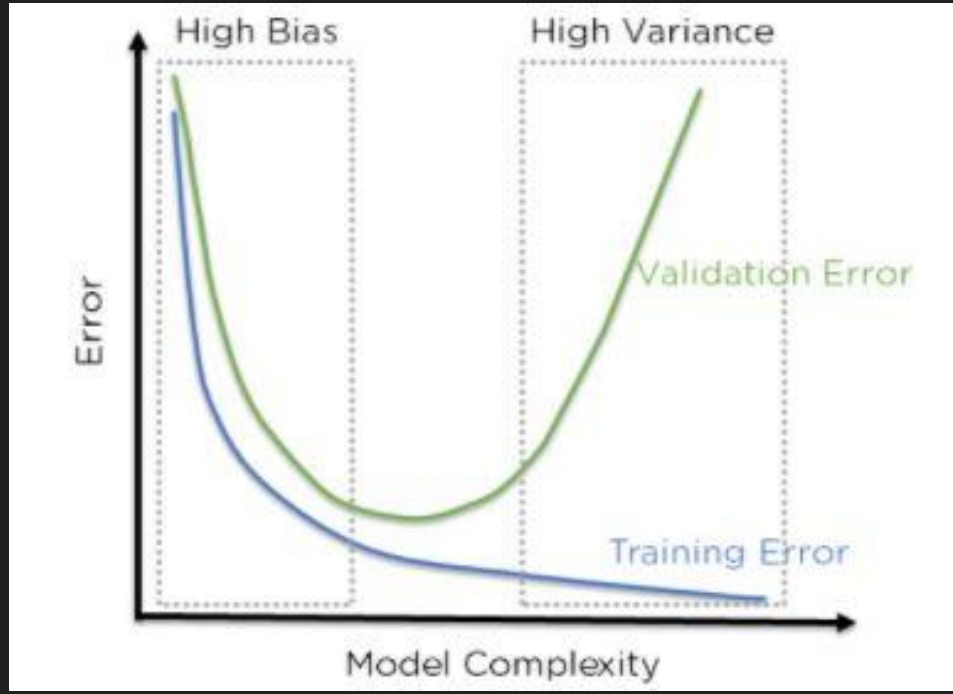


Methods to combat Underfitting

- Decreasing Regularization: By reducing the amount of regularization, the complexity and variability of the model increase, which facilitates better training.
- Increasing training time on the data
- Effective feature selection

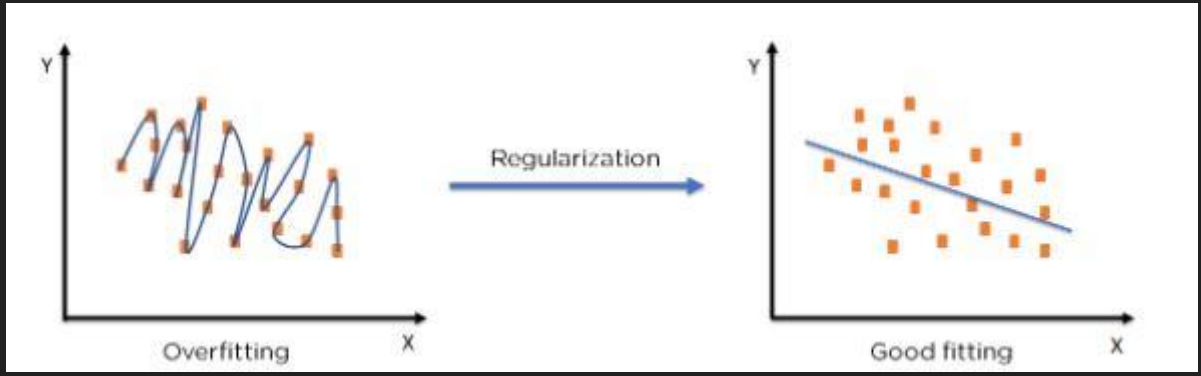
Bias and Variance

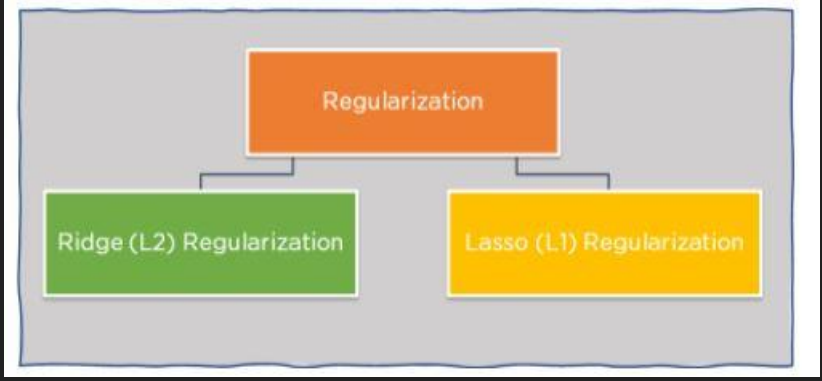
- Bias occurs when a machine learning model has limited flexibility for learning. Such a model pays excessive attention to the data and overly simplifies the rules. As a result, the error rate increases on both training and prediction data.
- Variance indicates the sensitivity of a machine learning model to the data. A model with high variance pays too much attention to the training data and fails to generalize well. Consequently, the prediction error is high. (In this case, the training error is low because it learns the training data too well, or it would be better to say, it learns excessively well.)
- Therefore, the levels of bias and variance must be carefully chosen to assist in the learning process, ensuring a balanced model that generalizes effectively.



Regularization

- This method imposes constraints on the machine learning model to prevent overfitting or underfitting.
- In regularization types L1 or L2, we can consider a penalty on the loss function to drive coefficients towards zero.
- L2 regularization allows weights to approach zero but not necessarily become zero, whereas in L1 regularization, weights can actually reach zero.





L1 Regularization or Lasso Regression or L1 norm

- Imagine we need an algorithm to predict an individual's exam rank based on their academic records. Naturally, not all features in a person's history have an equal impact on their rank. For instance, the person's GPA might weigh more heavily in the prediction than their extracurricular activities, or the average regional ranks might be more influential than their BMI. Therefore, with the help of L1 regularization during training, less influential features are assigned very small weights (close to zero) because their impact is minimal. For example, the weight related to regional average ranks tends towards a non-zero number, but the weight for BMI continually decreases towards zero.
- The red-colored part indicates the penalty term. Here, the sum of the absolute values of all weights is multiplied by a constant value (lambda). By adjusting lambda, the algorithm tries to minimize the cost function.

Cost function:

$$L(x, y) \equiv \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^{n} |\theta_i|$$

L2 Regularization or Ridge Regression or L2 norm

- In Ridge Regression, the cost function is derived from the sum of squared weights. In Ridge Regression, like Lasso, the algorithm aims to shrink the weights, but unlike Lasso, it does not force the weights to zero.
- When outliers are present in the dataset, this method does not perform well because the model's prediction error at outlier points can be very high, and with a penalty term like the following, the weights will be smaller:
- A machine learning model that uses Ridge performs better when all input features have a significant impact on predicting the target or output, and also, the weights inside the model are initialized equally

Cost function:

$$L(x,y) \equiv \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^{n} \theta_i^2$$

Comparison of L1 and L2

- L1 norm
- Adds the sum of the absolute values of weights to the cost/error.
- L1 Norm creates a sparser model.
- Depending on the effectiveness of different subsets of features, L1 Norm may create several different learnings.
- L1 Norm provides the feature selection property.
- L1 Norm performs better and is more robust when faced with outliers in input data.
- L1 Norm can create a simpler and more interpretable model, but this may prevent the algorithm from learning complex patterns.

- L2 norm
- Adds the sum of the squared values of weights to the cost/error.
- L2 Norm does not create a sparse model.
- L2 Norm only creates one learning/solution in the machine learning algorithm and does not act based on different subsets of features.
- L2 Norm does not provide the feature selection property.
- L2 Norm does not perform well when faced with outliers in input data.
- When the prediction target depends on all input features, L2 Norm leads to better learning.
- Unlike L1 Norm, L2 Norm can learn complex patterns in input data.

Combining the two previous methods

- Isn't it better to use both of them together to cover each other's weaknesses? Will it lead to better results? The answer is affirmative. Take a look at the Elastic Net regularization algorithm, which combines both L1 and L2 regularization methods!

The End

Thank you for your attention. I wish you pleasant times ahead.