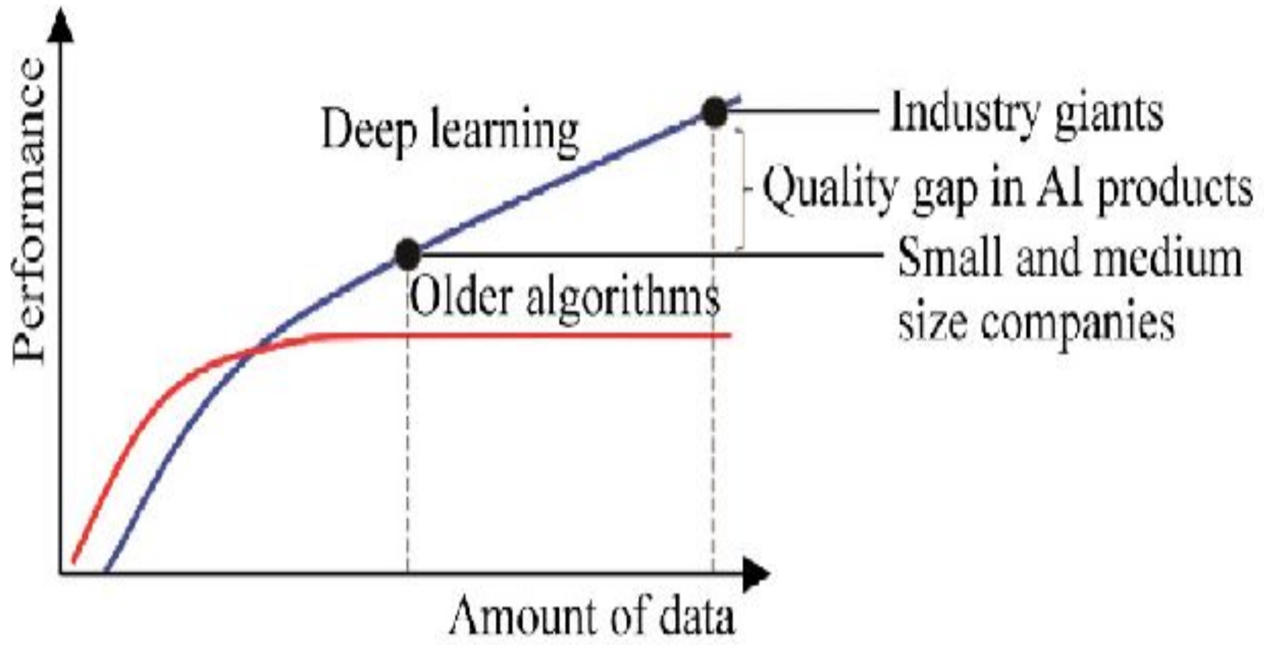
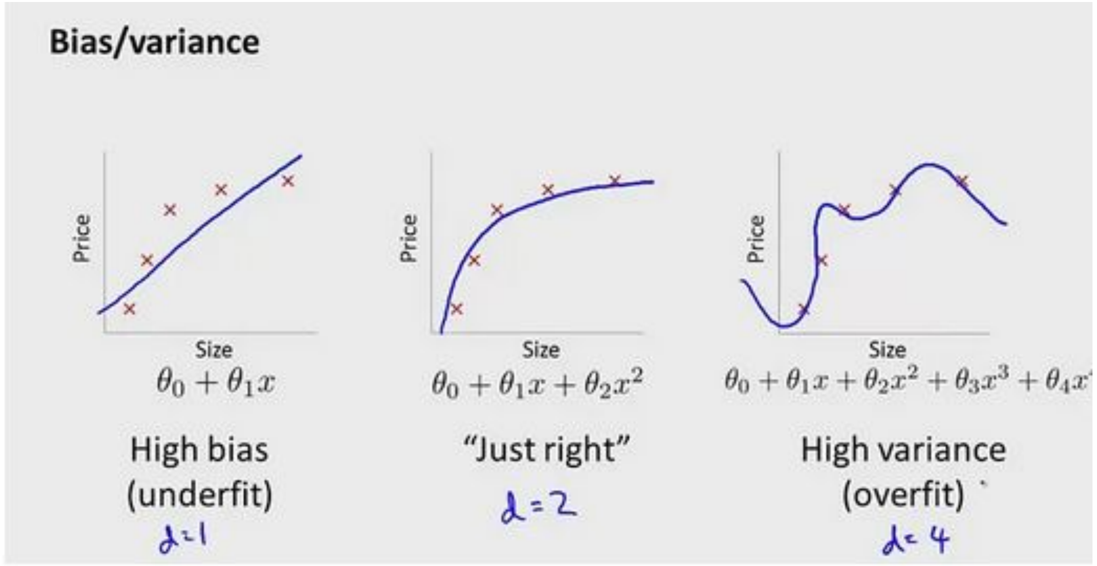
The Curse of small data-sets on Machine Learning

Plot showing the performance of ML with the increase in the data set size

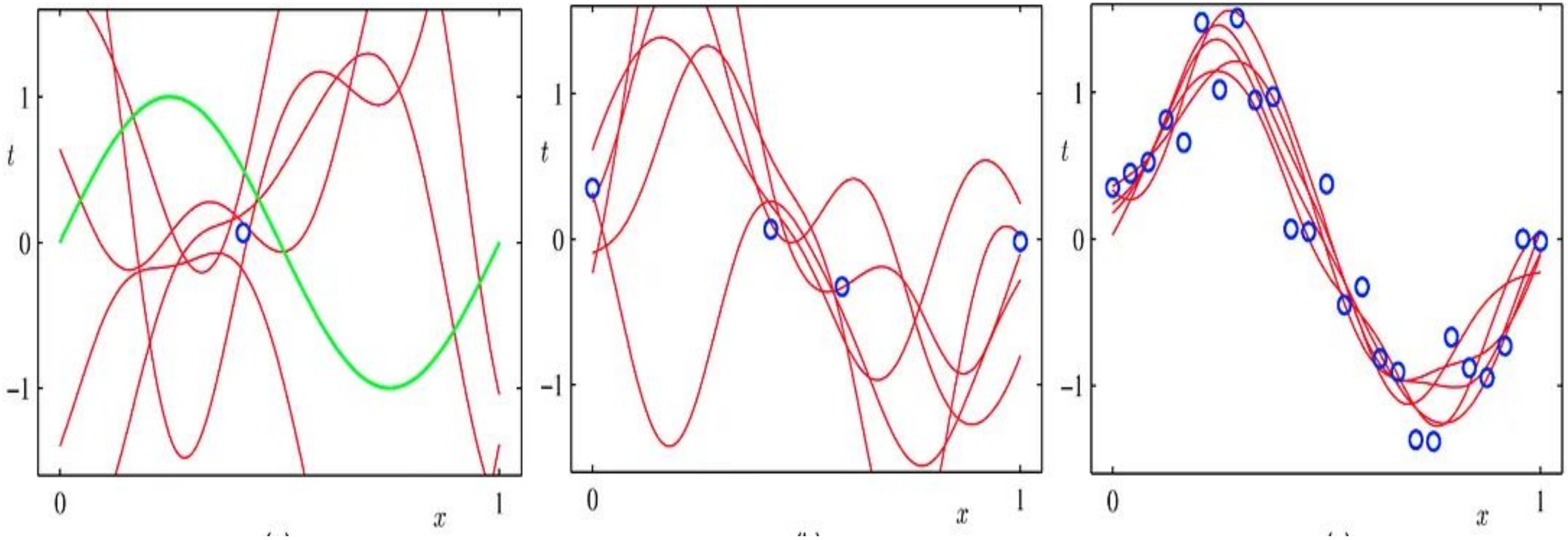


Bias and Variance



Bias signifies how under fitted our model is while variance signifies how overfitted the model is.
GOAL: Minimize Bias and Variance.

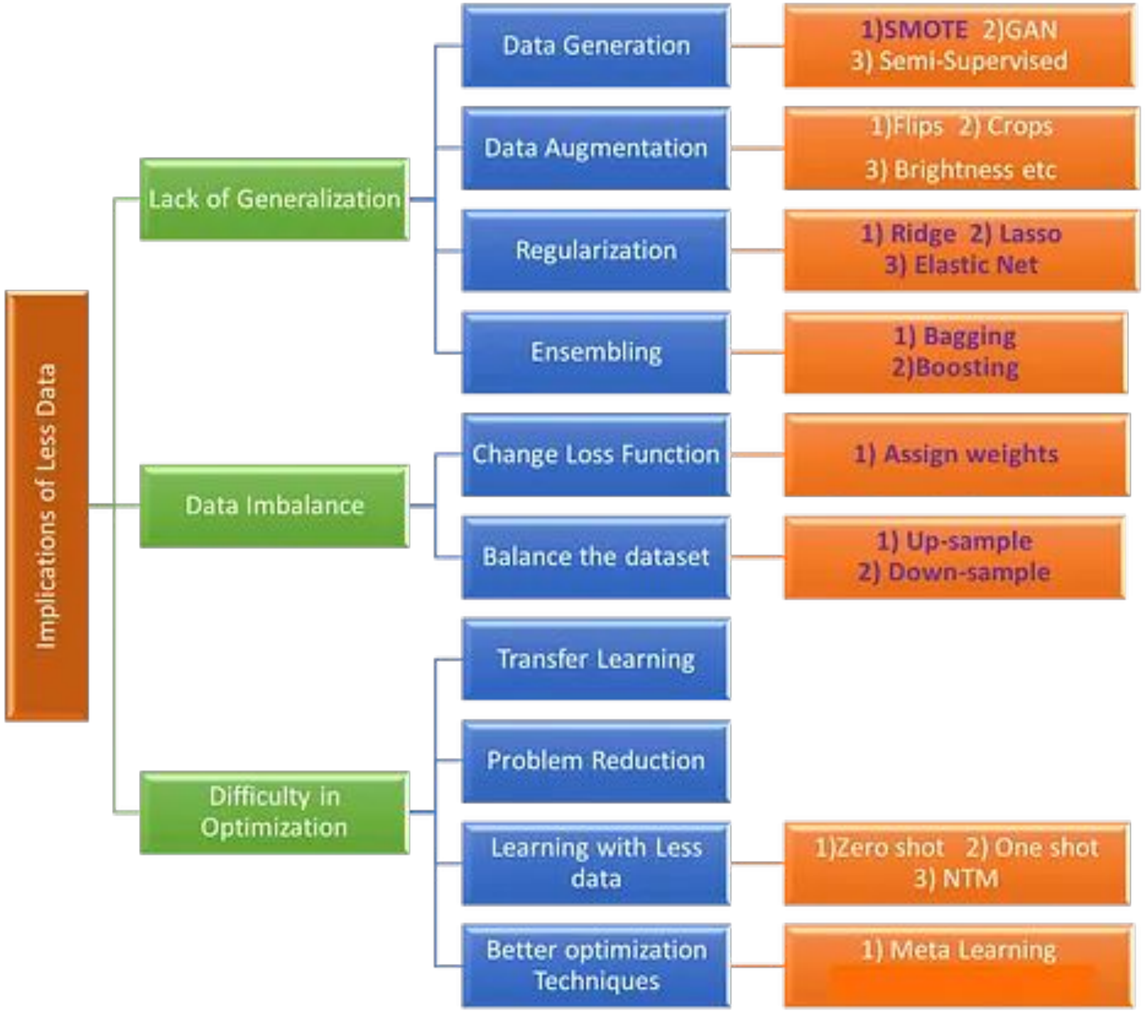
How to minimize Bias and variance?



Issues with small data

- **Over-fitting** becomes much harder to avoid
- Not only over-fit the training data, but sometimes validation set is over-fitted as well.
- **Outliers** become much more dangerous.
- Noise poses a real issue, be it in your target variable or in some of the features.

Possible solutions



Imbalanced Data-set

Since unbalanced data inherently penalizes majority class at different weight compared to a minority class, one solution to the problem is to make the data balanced.

Options to deal with imbalanced small datasets:

- Change loss function
- Up-sample or Down-sample

Up-sample

This can be done in case of small dataset by increasing the frequency of minority class.

Limitation: Making replicas would make the model overfit to the few examples.

Down-sample

This can be done by removing the examples from the majority class.

Limitation: Runs the risk of losing important information (undersample)

Change Loss Function

Cross-entropy loss ——–> Classification

Mean absolute error (MAE) and mean squared error (MSE) ——–> Regression

Problem: When dealing with unbalanced data, the model can become biased towards the majority class because it has a larger influence on the final loss value.

Solution: Weights can be added to the losses corresponding to different classes to balance out the influence of each class. Often referred to as "class weighting."

Example: If we have two classes with data in the ratio 4:1, we can apply weights in the ratio 1:4 to the loss function calculation to give the minority class more weight in the optimization process.

Anomaly Detection

Highly imbalanced datasets, such as those for fraud detection or machine failure, pose a challenge for traditional machine learning models.

The minority class is treated as an anomaly or outlier.

Identify instances that are significantly different from the majority of the data.

OneClassSVM: A type of support vector machine that learns to separate the majority class from the minority class by maximizing the margin around the minority class.

Clustering methods: such as k-means or DBSCAN, to group instances based on similarity and identify instances that are significantly different from their cluster.

Gaussian anomaly detection: Identifies instances that are far from the mean or have low probability under the distribution assuming that the data follows a normal (Gaussian) distribution.

Change Detection

Another related approach that aims to identify changes in behavior or patterns over time.

Can be implemented using time series analysis or other methods that compare the current state to a historical baseline and identify significant deviations.

Example: Change detection could be used to identify changes in user behavior, such as a sudden increase in website traffic or a change in purchasing patterns.

Drawback: Duplicate data increases the chances of oversampling

Ensembling Techniques

Aggregating multiple weak learners or different models can prove to be a great way to deal with imbalanced datasets.

Model aggregation techniques: Bagging and Boosting.

Bagging (Bootstrap Aggregating) involves training multiple models on different subsets of the data and combining their predictions to make a final prediction.

Boosting involves iteratively training models on the data, with each new model focusing on the examples that the previous models got wrong.

Both Bagging and Boosting have shown great results across a variety of problems, including imbalanced datasets.

Methods to overcome optimization problems

- **Transfer learning:**

  Technique in which an algorithm that has been trained to tackle a particular problem is slightly modified to tackle a similar problem. Example: in computer vision.

- **Problem reduction:**

  Technique to convert the modify the problem to one which has been solved using optimized algorithms. Example: voice clips to images in the case of audio classification

Learning with small datasets

One shot learning:

Used to refer to techniques to train machine algorithms with very small datasets

Ex: Siamese Network: One data point is taken as anchor point and a pair of similar and dissimilar data points are provided. The neural network tries to decrease the triple loss function.

$$L = max(||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + \alpha, 0)$$

Neural Turing Machine

It consists of controller and 2d memory bank. Reads and writes data from a 2d matrix called memory bank. It uses the data it receives from the outside world and the data present in the memory to make predictions. Detailed working:link

Reading:

$$r_i = \Sigma_i^k w_t(i) M_t(i)$$

Writing:

$$M_t = M_{t-1}(i)[1 - w_i(t)e_t + w_i(t)a_t]$$

Meta-learning

Finds the best way to learn from the given data. It tries to find the best hyperparameters for a model to learn from a particular dataset.

Consists of two models a high level model and low level model.

Low level model: Optimize or learner

High level model: Optimizer or meta-learner

Data augmentation

Generating new data by making slight changes to existing data.

Example:

In the case of images rotating, cropping, flipping and changing the brightness can be used to augment the data.

Data Generation

- SMOTE
- Semi-Supervised learning
- GAN

SMOTE (Synthetic Minority Over-sampling Technique)

Interpolates between existing minority class data points.

Selects a minority class data point and finds its k nearest neighbors in the feature space.

Randomly selects one of the neighbors and generates a new data point by interpolation

Limitation: Generation of noisy or unrealistic data points.

Modified- SMOTE

Distribution of features in the minority class and the density of minority class data points.

Data points more representative of the minority class.

Hyperparameter tuning is done on the specific dataset and problem.

Limitation: May not work for all data-set.

Semi-Supervised Learning

Solves the problem of having limited labeled data.

Builds a model that learns the pattern in the labeled data and uses this knowledge to predict the class of the unlabeled data, creating pseudo labels. Both labeled and pseudo-labeled data to train our model for the original task. Sets thresholds on predicted probabilities to select suitable pseudo-labeled data to train on.

Active learning : The model identifies which data points are most useful for training, and labels only a small set of data.

Co-teaching and Co-learning: Training multiple models simultaneously and exchanging information between them to improve performance. Used in situations where we have limited labeled data.

GAN (Generative Adversarial Networks)

Creates new data that looks very similar to real data.

Two components: a generator and a discriminator, which play against each other during training.

The generator creates fake data points, while the discriminator tries to distinguish between the real and fake data.

Example: If we want to generate new images of dogs, the generator creates fake dog photos, which are then fed to the discriminator along with real dog photos. The discriminator’s objective is to correctly identify the real and fake images, while the generator's objective is to create images that are indistinguishable from real ones.

Steps to be followed:

1. **Collect and preprocess the data:** Cleaning and normalizing the features.
2. **Choose a suitable regression algorithm:** Start with a simple regression algorithm like linear regression or polynomial regression. Experiment with more complex algorithms like support vector regression (SVR) or neural networks.
3. **Perform feature selection:** Principal component analysis (PCA) or Lasso regression to identify the most important features in your data. Reduces overfitting and improves the model's accuracy.
4. **Split the data into training and testing sets:** Cross-validation techniques like k-fold cross-validation to split your data into training and testing sets. Evaluate the model's performance and avoid overfitting.
5. **Regularize the model:** Using regularization techniques like L1 or L2 regularization to prevent overfitting and improve your model's accuracy.
6. **Train and test the model:** Using metrics like mean squared error (MSE), root mean squared error (RMSE), and R-squared to evaluate your model's performance.