VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

HUỲNH TRẦN MINH TIẾN

NGUYỄN TRUNG TÍN

MIDTERM ESSAY

MIDTERM PROJECT

INTRODUCTION TO MACHINE LEARNING

HO CHI MINH CITY, YEAR 2023

VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

HUỲNH TRẦN MINH TIẾN

NGUYỄN TRUNG TÍN

MIDTERM ESSAY

MIDTERM PROJECT

INTRODUCTION TO MACHINE LEARNING

Advised by
Prof., Dr. Le Anh Cuong

HO CHI MINH CITY, YEAR 2023

ACKNOWLEDGMENT

In order to get a complete and good report like this, our team has received enthusiastic help from lectures and classmates.

My team would also like to thank for the helpful knowledge and enthusiastic help from lecturer Prof. Le Anh Cuong. Thank you, Prof. Cường, for being so enthusiastic in teaching, educating, equipping us with the necessary knowledge and creating the most favorable conditions for us to complete this report.

And also thank Ton Duc Thang University for giving us a modern and developed educational environment.

With hard work and effort we have successfully completed this report. But surely, this report cannot avoid mistakes. We are looking forward to receiving from teacher so that we can improve it better.

We sincerely thank you!

Ho Chi Minh City, October 23 year 2023

Author

(signature and full name)

Huynh Tran Minh Tien

Nguyen Trung Tin

DECLARATION OF AUTHORSHIP

We hereby declare that this thesis was carried out by ourselves under the guidance and supervision of Prof. Le Anh Cuong; and that the work and the results contained in it are original and have not been submitted anywhere for any previous purposes. The data and figures presented in this thesis are for analysis, comments, and evaluations from various resources by my own work and have been duly acknowledged in the reference part.

In addition, other comments, reviews and data used by other authors, and organizations have been acknowledged, and explicitly cited.

We will take full responsibility for any fraud detected in my thesis. Ton Duc Thang University is unrelated to any copyright infringement caused on my work (if any)

Ho Chi Minh City, October 23 year 2023

Author

(signature and full name)

Huynh Tran Minh Tien

Nguyen Trung Tin

EVALUATION OF INSTRUCTING LECTURER

Confirmation of the instructor

_____
_____
_____
_____
_____
_____
_____

Ho Chi Minh City, 2023

(sign and write full name)

The assessment of the teacher marked

_____
_____
_____
_____
_____
_____
_____

Ho Chi Minh City, 2023

(sign and write full name)

ABSTRACT

Machine learning has emerged as a ubiquitous force, permeating various facets of our lives. Its prowess stems from its ability to learn from data, empowering computers to discern patterns and make predictions without explicit programming. However, venturing into the realm of machine learning unveils a critical phenomenon known as overfitting. This occurs when a model becomes overly reliant on the intricacies of the training data, impairing its ability to generalize to unseen data.

This essay embarks on a journey into the labyrinth of machine learning, delving into the depths of fundamental models such as k-Nearest Neighbors (kNN), linear regression, Naive Bayes classifiers, and decision trees. Each model possesses unique strengths and limitations, contributing to the vast tapestry of machine learning.

The essay then delves into the intricacies of overfitting, exploring its various manifestations and detrimental effects on model performance. Through meticulous examination, the essay unveils a range of solutions aimed at mitigating the perils of overfitting. Regularization techniques, early stopping, and ensemble learning emerge as potent weapons in the battle against overfitting.

In conclusion, the essay presents a comprehensive overview of machine learning models, overfitting, and solutions. By gaining a deeper understanding of these concepts, practitioners can navigate the labyrinth of machine learning with greater confidence, harnessing its immense power to solve real-world problems.

TABLE OF CONTENTS

LIST OF TABLES, PICTURES, GRAPHS

LIST OF TABLES

LIST OF FIGURES

Question 1:

AI is a technique that makes the computer system smarter and simulates a human for solving complex tasks. Otherwise, ML is a subset of AI that allows a machine to automatically learn from past data without programming explicitly and give back the accurate output. The ML models are tools that develop computer programs that can learn and make predictions or decisions. The models are created from algorithms, which are trained using labeled, unlabeled, or mixed data.

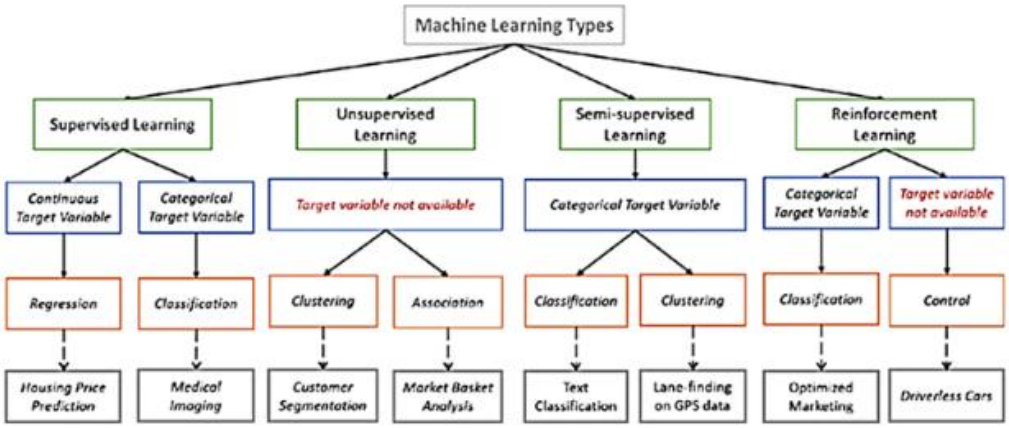

Figure 1: Machine Learning Types

There are four main types of ML Algorithm: Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning.

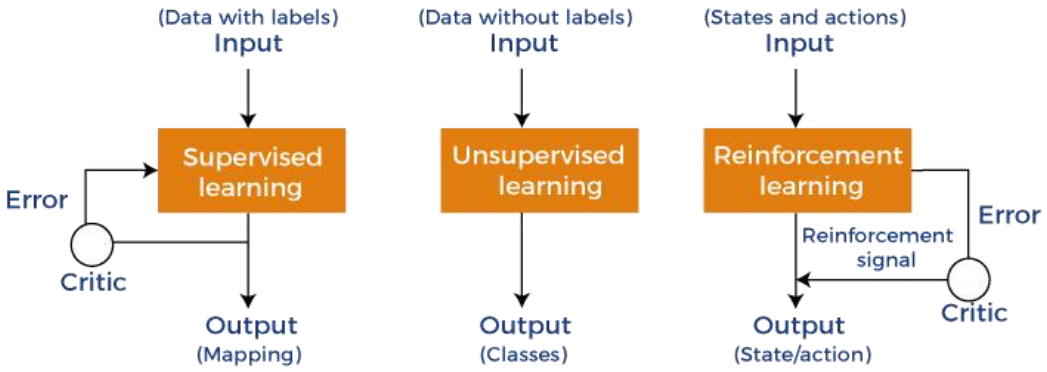

Figure 2: Main Types of ML Algorithm

1. Supervised Learning

This category is a machine learning approach where algorithms learn from labeled datasets to train algorithms that accurately classify data or predict outcomes.

When an algorithm is trained on labeled data, it learns to recognize patterns in the input data and correlate them with the proper output. The labeled data gives the algorithm a set of instances from which to learn, with each example consisting of an input and a corresponding output. The algorithm then utilizes these examples to learn a mapping between inputs and outputs that can be used to make predictions on fresh, previously unseen data.

During training, the algorithm adjusts its internal parameters or weights to minimize the gap between its predicted output and the true output for each example in the training set. This is called as optimization or fitting.

The goal is to find a set of weights that can be applied to new, previously unknown data. Once trained on labeled data, the system can be used to generate predictions on new, unlabeled data. Based on the mapping it learned during training, the algorithm takes an input and generates an output. Supervised learning assists enterprises in solving a wide range of real-world challenges at scale.

It is used to solve problems with data-mining *classification* and *regression*:

1) Regression is a statistical method for determining the relationship between dependent and independent variables. It is widely used to produce forecasts, such as those for a company's sales revenue. Regression is a type of supervised learning problem where the goal is to predict a continuous numerical value. For example, we might want to predict the price of a house based on its size, location, and other features. Linear regression is a common algorithm used for regression problems.

2) Classification is another type of supervised learning problem where the goal is to predict a categorical value. An algorithm is used in classification to accurately allocate test data to certain categories. It identifies specific entities within the dataset and tries to derive conclusions about how those items should

be labeled or described. For example, we might want to classify an email as spam or not spam based on its content. Logistic regression and decision trees are common algorithms used for classification problems.

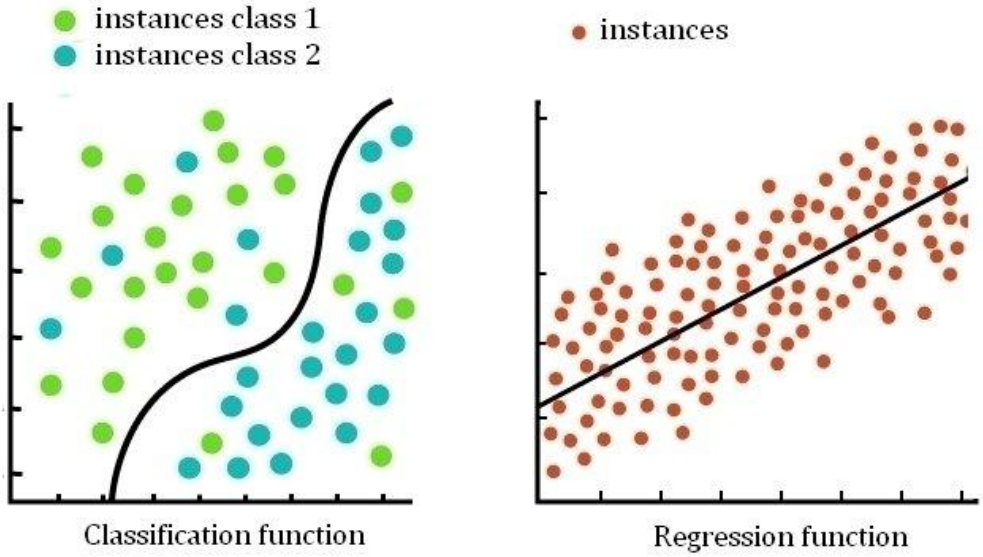

Figure 3: Classification and Regression Function

2. Unsupervised Learning

The method in unsupervised machine learning is trained on an unlabeled dateset, which means there are no predefined labels or classifications for the data. The purpose of unsupervised learning is to find patterns or groupings in data without knowing what those patterns or groupings are.

Unsupervised learning can be used for a range of tasks, including clustering similar data points together, recognizing errors or outliers in the data, and reduce the dimensionality of the data.

1) Clustering

Clustering is an unsupervised machine learning process that includes clustering data points with similar features. It is used to find patterns and relationships in unlabeled or unclassified data. Clustering algorithms aggregate unstructured data based on similarities and patterns.

In machine learning, clustering is an important topic. It saves time for data analysts by delivering algorithms that improve data categorization and investigation. It is

also significant in well-defined network models. Because of rapid data changes and a scarcity of labels, many analysts favor unsupervised learning in network traffic analysis (NTA). It is required while developing improved forecasting, particularly in the area of threat detection. This can be accomplished by creating network logs that improve threat visibility.

Figure 4: Clustering Algorithm

2) Association Rule Learning

Association rule learning is an unsupervised learning technique that seeks for interesting associations between variables in a dataset. It is used to find patterns in massive datasets by examining the frequency of item co-occurrence. The method is based on the concept of association rules, which are used to find links between variables that occur together frequently. Association rules are formed by evaluating data and detecting frequent itemsets, which are groups of items that appear frequently together. The Apriori algorithm is the most commonly used algorithm for this.

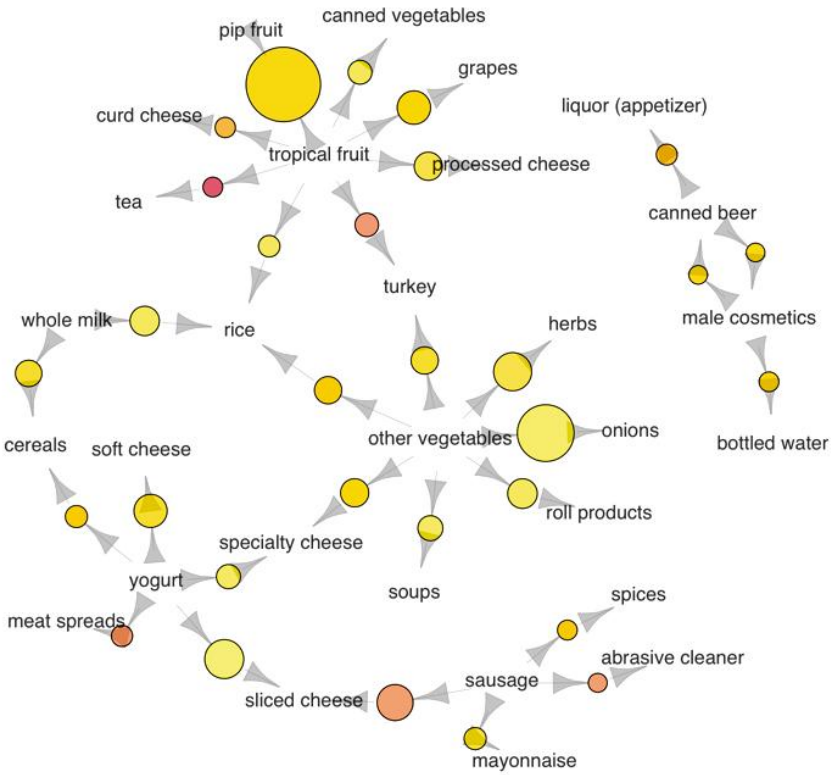

Figure 5:Association Rule

3) Dimentionality Reduction

Dimensionality reduction is an unsupervised learning that reduces the number of features in a dataset while maintaining as much of the critical information as possible. It is used to solve the curse of dimensionality, which is a common issue in machine learning in which the model's performance declines as the number of features rises.

Dimensionality reduction can be approached in two ways: feature selection and feature extraction. The process of picking a subset of the original features that are most relevant to the problem at hand is known as feature selection. The goal is to minimize the dataset's dimensionality while maintaining the most significant attributes. There are numerous methods for selecting features, such as filter methods, wrapper methods, and embedding methods. Feature extraction, on the other hand, entails developing new features by combining or altering the existing features.

3. Semi-Supervised *Learning*

Semi-supervised machine learning is a type of machine learning that trains a model using both labeled and unlabeled data. It's valuable when there's a lot of unlabeled data yet classifying it all would be too expensive or difficult. Semi-supervised learning can increase model performance by capturing the shape of the underlying data distribution using unlabeled data.

Semi-supervised learning examples include:

1. *Text classification*: Using a small quantity of labeled data and a big amount of unlabeled text data, a model can be trained to classify text documents into categories. For example, a model can be trained to categorize news articles into sports, politics, entertainment, and so on.
2. *Image classification*: Using a small quantity of labeled data and a big amount of unlabeled image data, a model may be trained to categorize photos into categories. A model, for example, can be trained to recognize faces, animals, objects, and so on.
3. *Anomaly detection*: Using a little quantity of labeled data and a large amount of unlabeled data, a model can be trained to find strange or abnormal patterns or observations in the data. A model, for example, can be trained to detect fraud, malware, outliers,etc.

4. Reinforcement *Learning*

Reinforcement learning is a sort of machine learning in which an agent interacts with its environment to learn how to make a series of decisions that maximizes a cumulative reward. The agent learns by obtaining feedback in the form of rewards or penalties for each action it takes. The goal is to learn a policy that maps states to actions that maximize the predicted cumulative reward.

Along with supervised and unsupervised learning, reinforcement learning is one of the four major types of machine learning. It varies from supervised learning in that it does not require labeled input/output pairs to be provided, and it differs from unsupervised learning in that it requires an agent interacting with an environment to learn how to make decisions.

5. Comparison *Models*

	kNN	Linear Regression	Naive Bayes Classifier	Decision Tree
Types of Model	Non-parametric	Parametric		
Goals	To classify or predict new data points	To predict a continuous value for a new data points	To classify new data points	To classify or predict new data points
Learning Criteria	- The model classifies individual point by a majority voting of its neighbors, measured by a Distance metrics, such as Euclidean distance, Manhattan distance, or cosine similarity. - Minimize the distance between the	- The model finds the best fits line (equation) for predicting the target variable for new data points. - Minimizing the sum of squared errors (SSE) between predicted and the actual value,	- The model is a probabilistic classification algorithm which uses Bayes's theorem which calculate the probability of an event happening to classify data points. - The learning criterion is to	- The model works by recursively partitioning the data into smaller subsets until each subset contains only data points from a single class. - The learning criterion is to maximize the information gain at each

	new data point and its K nearest neighbors.		maximize the posterior probability of the class, given the predictor variable.	split which is a measure of how much purity of the data improved by splitting.
Data	- Categorical, and numerical	- Numerical	- Categorical, and numerical	- Categorical, and numerical
Types of Problems	Classification	Regression	Classification	Classification & Regression
Strength	- Simple to understand and implement. - Can be used for both classification and Regression. - Robust to outliers.	- Simple to understand and interpret. - Can be used for both classification and Regression. - Efficient to train and predict.	- Simple to understand and implement. - Can be used for both classification and Regression. - Efficient to train and predict.	- Complex relationships between predictor and target variables. - Can be used for both classification and Regression.
Weakness	Computationally expensive for large datasets.	Sensitive with outliers in case the relationship	Sensitive with outliers in case the predictor	Can be overfit to the training data.

	Sensitive to choice of distance metric and the value of k.	between predictor and the target variables is linear.	variables are independent.	Sensitive to outliers.

Table 1: Comparision Models Table

Question 2:

Describe: A machine learning problem using a cancer patient dataset is a problem that uses machine learning models to analyze data about cancer patients. This data can include information about patients such as:

- Demographic information (age, sex, race, ethnicity)
- Medical history (previous diagnoses, medications, surgeries)
- Lifestyle factors (smoking, alcohol consumption, diet, exercise)
- Biomarker data (genetic mutations, protein expression levels)
- Imaging data (X-rays, CT scans, MRIs)
- Clinical outcome data (tumor size, stage, grade, survival)

The machine learning models used in these problems can be classified into two main types:

- Classification models are used to predict whether a patient has cancer or not.
- Regression models are used to predict features of the tumor, such as size, stage, or survival time.

Examples of machine learning problems using a cancer patient dataset:

- Cancer risk prediction: This model can be used to predict whether a person has a high or low risk of developing cancer based on factors such as age, sex, family history, and lifestyle.
- Cancer diagnosis: This model can be used to diagnose cancer by analyzing imaging or biomarker data.
- Cancer classification: This model can be used to classify cancer into different types, such as breast cancer, lung cancer, or colorectal cancer.

- Treatment outcome prediction: This model can be used to predict the outcome of a particular treatment for a cancer patient.

Question 3:

1. Knowledge section

- Noise: is meaningless or irrelevant data present in the dataset. It affects the performance of the model if it is not removed.
- Bias: Bias is a prediction error that is introduced in the model due to oversimplifying the machine learning algorithms. Or it is the difference between the predicted values and the actual values.
- Variance: If the machine learning model performs well with the training dataset, but does not perform well with the test dataset, then variance occurs.
- Generalization: It shows how well a model is trained to predict unseen data.

2. What is a good fitting model?

Good Fitting is a crucial aspect that needs to be achieved in order to obtain optimal results in problem-solving. The objective of achieving Good Fitting can be quite challenging to implement in reality, as it necessitates close monitoring of the machine learning algorithm's performance over time while it undergoes the learning process on the training data set. In order to assess the level of Good Fitting, it becomes imperative to delve into the intricacies of the model parameters and accuracy, both in relation to the training data set and the new data sets.

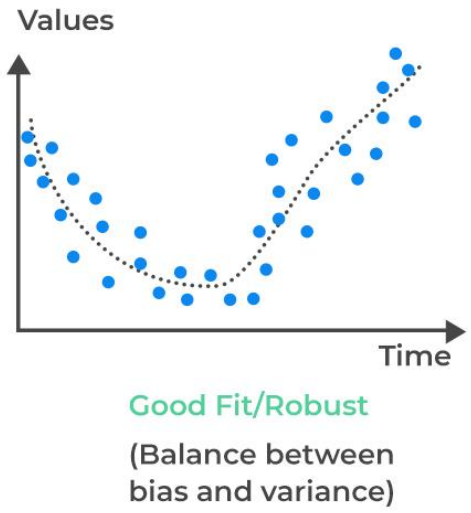

Figure 6: Good Fitting Model

As the learning process unfolds and progresses, it is expected that the model's error on the training data set will gradually decrease. However, it is important to note that if the training process is prolonged excessively, the model's accuracy might suffer adverse effects due to the emergence of the Overfitting problem. This occurs when the learning process incorporates noisy and anomalous data from the training set, thus leading to a decrease in the model's generalizability. Consequently, the error associated with the validation data set will witness an increase, as the model's ability to generalize diminishes.

3. What is a underfitting model?

When a model has not learned the patterns in the training data well and is unable to generalize well on the new data, it is known as underfitting. An underfit model has poor performance on the training data and will result in unreliable predictions.

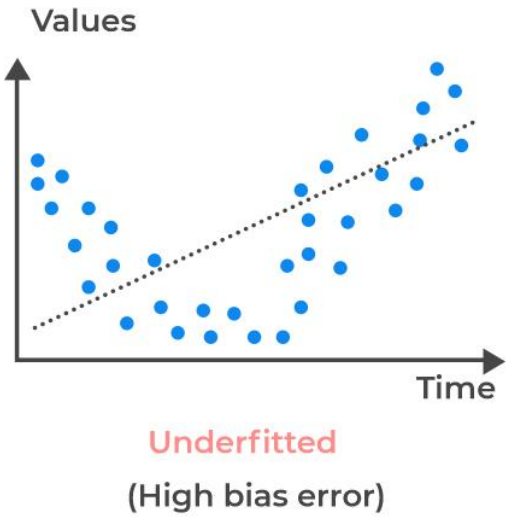

Figure 7: Unfitting Model

a. Reasons for Underfitting

- Data used for training is not cleaned and contains noise (garbage values) in it
- The model has a high bias
- The size of the training dataset used is not enough
- The model is too simple

b. Ways to Tackle Underfitting

- Increase the number of features in the dataset
- Increase model complexity
- Reduce noise in the data
- Increase the duration of training the data

4. Overfitting definition

Over-fitting presents a significant challenge in the realm of supervised machine learning tasks. It is a phenomenon that arises when a learning algorithm impeccably fits the training data set, resulting in the memorization of not only the noise but also the idiosyncrasies present in the training data.

When data scientists use machine learning models for making predictions, they first train the model on a known data set. Then, based on this information, the model

tries to predict outcomes for new data sets. The outcome of an overfit model can give inaccurate predictions and cannot perform well for new data set. (unknown data set).

EX:

a use case where a machine learning model has to analyze photos and identify the ones that contain dogs in them. If the machine learning model was trained on a data set that contained majority photos showing dogs outside in parks , it may may learn to use grass as a feature for classification, and may not recognize a dog inside a room.

Figure 8: Example of Overfitting

The amount of data used for learning process is fundamental in this context. Small data sets are more prone to over-fitting than large data sets, and despite the complexity of some learning problem, large data sets can even be affected by over-fitting. Overfitting of the training data leads to deterioration of generalization properties of the model, and results in its untrustworthy performance when applied to novel measurements.

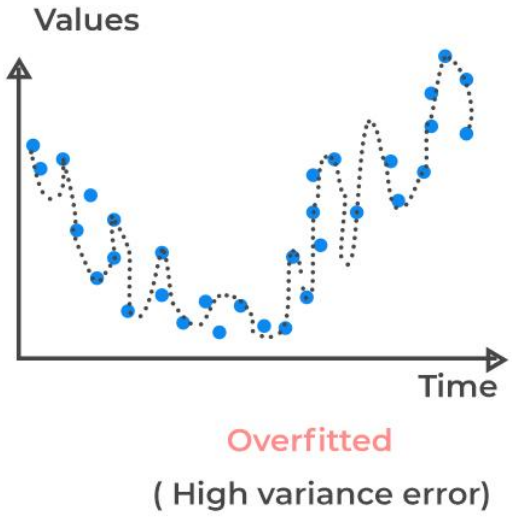

Figure 9: Overfitting Model

5. Underfitting vs Overfitting

Underfit models experience high bias and low variance - they give inaccurate results for both the training data and test set. On the other hand, overfit models experience high variance and low bias - in many cases, small increases in bias result in large decreases in variance - they give accurate results for the training set but not for the test set. More model training results in less bias but variance can increase. Data scientists aim to find the sweet spot between underfitting and overfitting when fitting a model. A well-fitted model can quickly establish the dominant trend for seen and unseen data sets.

When the model under-fits, the bias is generally high and the variance is low. Over-fitting is typically characterized by high variance, low bias estimators. In many cases, small increases in bias result in large decreases in variance

EX: There are 50 data points generated using a cubic polynomial plus noise. This data set is divided into two, 30 red data points for training data, 20 yellow data points for test data. The graph of this cubic polynomial is given by the green line. Our problem is, assuming we do not know the initial model but only the data points, find a “good” model to describe the given data.

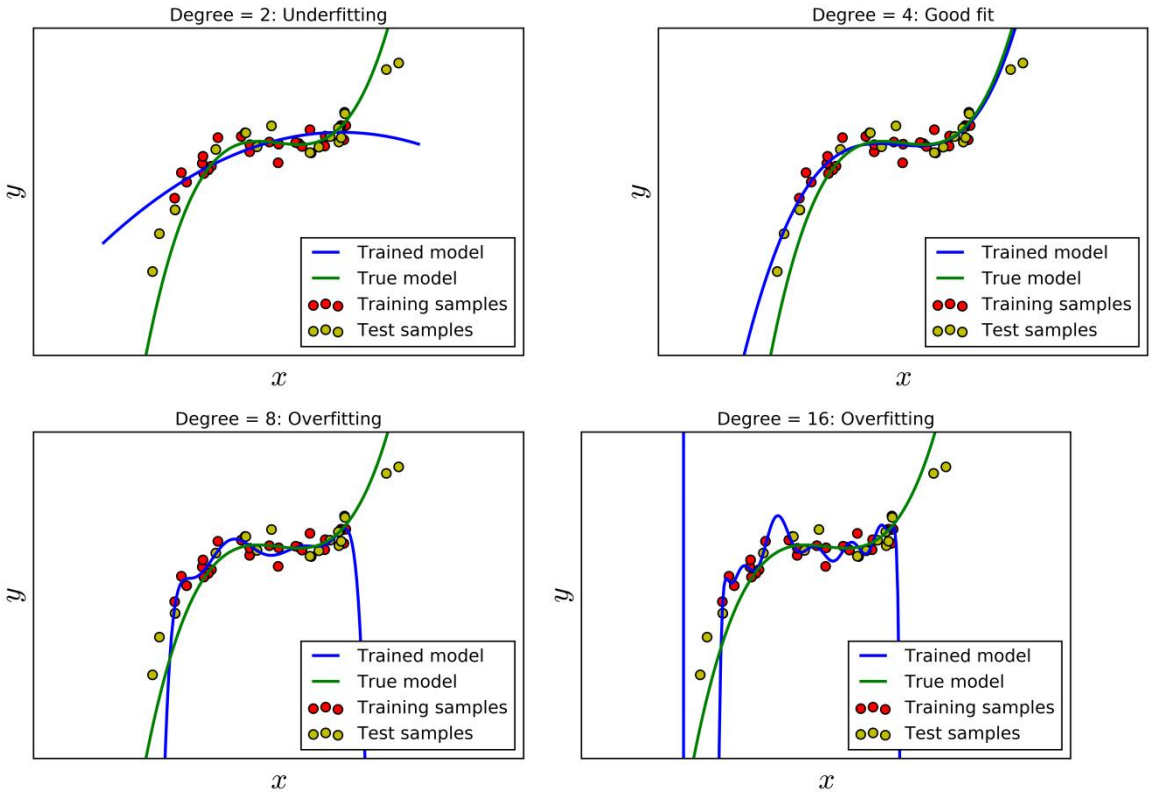

Figure 10: Overfitting example

It is clear that a polynomial of degree not exceeding 29 can fit perfectly with 30 points in the training data. Let's consider some values d=2,4,8,16.

With d=2, the model is not really good because the predicted model is too different from the actual model. In this case, we say the model is underfitting .

With d=4, we get the predicted model quite similar to the real model. The highest degree coefficient found is very close to 0 , so this 4th degree polynomial is quite close to the original 3rd degree polynomial. This is a good model.

With d=8, with data points in the range of training data, the predicted model and the real model are quite similar. However, on the right side, the 8th degree polynomial gives results completely opposite to the trend of the data . The same thing happens in cased=16. This 16th degree polynomial is too tight to fit the data in the range under consideration, and is too tight , meaning it is not smooth in the range of training data. Overfitting in the 16th order case is not good because the model is trying to describe the noise rather than the data. These two cases of higher degree polynomials are called Overfitting.

6. Reasons for Overfitting

- High variance and low bias, your training accuracy increases, but validation accuracy decreases with the number of epochs.
- Dataset has noisy data (not clean) or inaccurate points (garbage values): it may decrease the validation accuracy and increase the variance.
- Too complex: the variance will rise, and the bias will be low. It can learn too much noise or random fluctuations using the training data, which hinders the performance of data the model has never seen before.
- If the size of the training dataset is not enough, then the model will get to explore only some of the scenarios or possibilities. When introduced to unseen data, the accuracy of the prediction will be less.

7. Way to detect overfitting?

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.

During each iteration, the steps are:

1. Keep one subset as the validation data and train the machine learning model on the remaining K-1 subsets.
2. Observe how the model performs on the validation sample.
3. Score model performance based on output data quality.

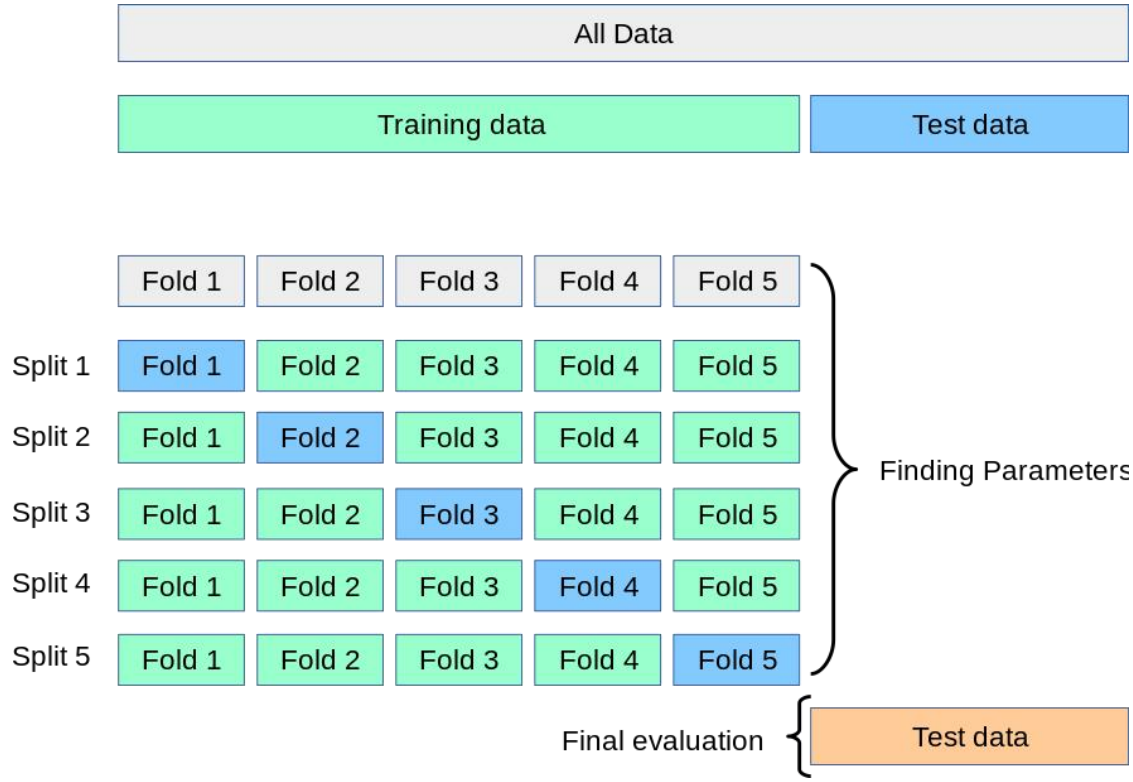

Figure 11: Cross Validation

8. Way to avoid overfitting

a. Train with more data

Expanding the training set to include more data can increase the accuracy of the model by providing more opportunities to parse out the dominant relationship among the input and output variables.

It may not always work to prevent overfitting, but this way helps the algorithm to detect the signal better to minimize the errors.

When a model is fed with more training data, it will be unable to overfit all the samples of data and forced to generalize well.

But in some cases, the additional data may add more noise to the model; hence we need to be sure that data is clean and free from in-consistencies before feeding it to the model.

b. Data augmentation

Data Augmentation is a data analysis technique, which is an alternative to adding more data to prevent overfitting. In this technique, instead of adding more training data, you can apply various transformations to the existing dataset or add a slightly modified copies of already existing data to increase the dataset size artificially. Augmentation is a common technique to increase the sample size of data for a model, particularly in computer vision.

EX: applying transformations such as translation, flipping, and rotation to input images.

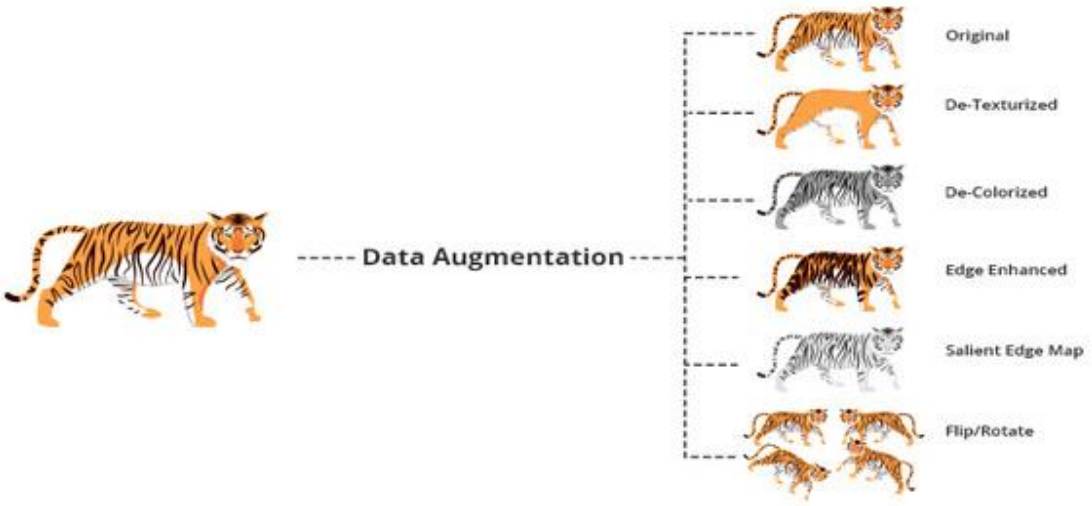

Figure 12: Data Augmentation

c. Pruning (Feature selection)

While building the ML model, we have a number of parameters or features that are used to predict the outcome. However, sometimes some of these features are redundant or less important for the prediction, and for this feature selection process is applied. In the feature selection process, we identify the most important features within training data, and other features are removed. Further, this process helps to simplify the model and reduces noise from the data (this is commonly mistaken for dimensionality reduction, but it is different). Some algorithms have the auto-feature selection, and if not, then we can manually perform this process.

Some feature selection heuristics:

- ✓ Variance Thresholds
- ✓ Correlation Thresholds
- ✓ Genetic Algorithms (GA)
- ✓ Honorable Mention: Stepwise Search

EX: To predict if an image is an animal or human, you can look at various input parameters like face shape, ear position, body structure, etc. You may prioritize face shape and ignore the shape of the eyes.

Figure 13: Example of Feature Selection

d. Early stopping

In this technique, the training is paused before the model starts learning the noise within the model. In this process, you can do this by training multiple times and figuring out roughly at what point noise starts to impact your training. Continue up to a certain number of iterations until a new iteration improves the performance of the model. Your training graphs will help inform the optimal time to stop training.

However, this technique may lead to the underfitting problem if training is paused too early. So, getting the timing right is important to find that "sweet spot" between underfitting and overfitting. The final parameters returned will enable the model to have low variance and better generalization. The model at the time the training is stopped will have a better generalization performance than the model with the least training error.

Early stopping can be thought of as implicit regularization, contrary to regularization via weight decay. This method is also efficient since it requires less

amount of training data, which is not always available. Due to this fact, early stopping requires lesser time for training compared to other regularization methods. Repeating the early stopping process many times may result in the model overfitting the validation dataset, just as similar as overfitting occurs in the case of training data. The number of iterations(i.e. epoch) taken to train the model can be considered a hyperparameter. Then the model has to find an optimum value for this hyperparameter (by hyperparameter tuning) for the best performance of the learning model.

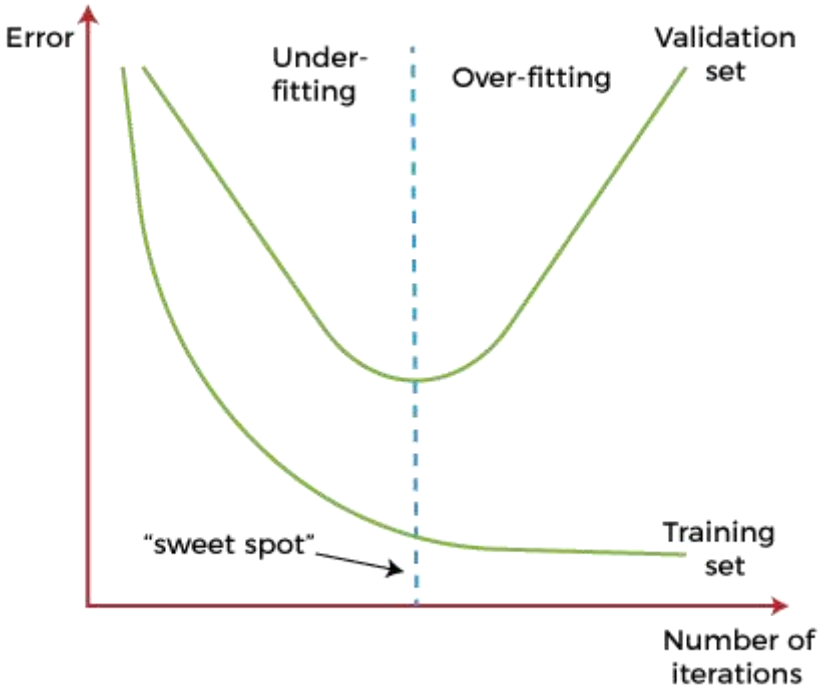

Figure 14: Early Stopping

e. Regularization

If the phenomenon of overfitting arises as a consequence of excessive complexity within a model, it would be logical for us to mitigate this issue by diminishing the quantity of features present. However, in situations where our understanding is inadequate for determining the inputs to be eliminated during the feature selection process,regularization techniques can prove to be exceedingly advantageous. The implementation of the regularization technique may result in a slight increase in

bias but concurrently leads to a slight reduction in variance. Within this technique, we modify the objective function by incorporating a penalizing term, which possesses a higher value in the presence of a more intricate model.

Types of regularization techniques:

- **Ridge Regression**

Ridge regression is one of the types of linear regression in which a small amount of bias is introduced so that we can get better long-term predictions.

In this technique, the cost function is altered by adding the penalty term to it. The amount of bias added to the model is called Ridge Regression penalty. We can calculate it by multiplying with the lambda to the squared weight of each individual feature.

The equation for the cost function in ridge regression will be:

$$\sum_{i=1}^{M}(y_i - y_i^{'})^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{n}\beta_j * x_{ij} \right)^2 + \lambda \sum_{j=0}^{n}\beta_j{}^2$$

In the above equation, the penalty term regularizes the coefficients of the model, and hence ridge regression reduces the amplitudes of the coefficients that decreases the complexity of the model.

As we can see from the above equation, if the values of λ tend to zero, the equation becomes the cost function of the linear regression model. Hence, for the minimum value of λ, the model will resemble the linear regression model.

A general linear or polynomial regression will fail if there is high collinearity between the independent variables, so to solve such problems, Ridge regression can be used.

It helps to solve the problems if we have more parameters than samples.

- **Lasso Regression**

Lasso regression is another regularization technique to reduce the complexity of the model. It stands for Least Absolute and Selection Operator.

It is similar to the Ridge Regression except that the penalty term contains only the absolute weights instead of a square of weights.

Since it takes absolute values, hence, it can shrink the slope to 0, whereas Ridge Regression can only shrink it near to 0.

It is also called as L1 regularization. The equation for the cost function of Lasso regression will be:

$$\sum_{i=1}^{M}(y_i - y_i^{'})^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{n}\beta_j * x_{ij}\right)^2 + \lambda\sum_{j=0}^{n}|\beta_j|$$

Some of the features in this technique are completely neglected for model evaluation.

Hence, the Lasso regression can help us to reduce the overfitting in the model as well as the feature selection.

Key Difference between Ridge Regression and Lasso Regression

- ✧ Ridge regression is mostly used to reduce the overfitting in the model, and it includes all the features present in the model. It reduces the complexity of the model by shrinking the coefficients.
- ✧ Lasso regression helps to reduce the overfitting in the model as well as feature selection.

f. Ensemble methods

Ensembles are machine learning methods for combining predictions from multiple separate models. There are a few different methods for ensembling, but the two most common are:

Bagging: attempts to reduce the chance overfitting complex models.

- It trains a large number of "strong" learners in parallel.
- A strong learner is a model that's relatively unconstrained.
- Bagging then combines all the strong learners together in order to "smooth out" their predictions.

Boosting: attempts to improve the predictive flexibility of simple models.

- It trains a large number of "weak" learners in sequence.
- A weak learner is a constrained model (i.e. you could limit the max depth of each decision tree).
- Each one in the sequence focuses on learning from the mistakes of the one before it.
- Boosting then combines all the weak learners into a single strong learner.

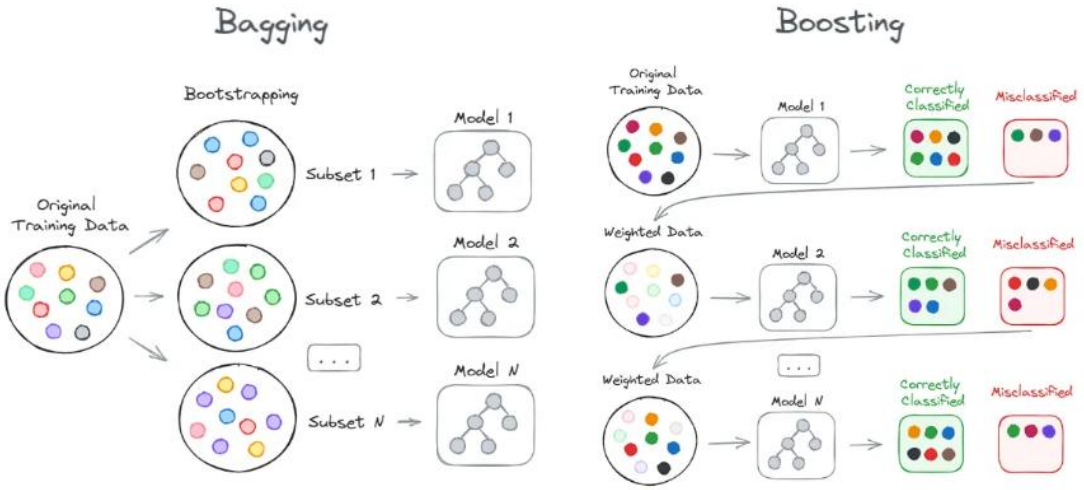

Figure 15:Bagging and Boosting Methods

While bagging and boosting are both ensemble methods, they approach the problem from opposite directions.

Bagging uses complex base models and tries to “smooth out” their predictions, while boosting uses simple base models and tries to “boost” their aggregate complexity.

REFERENCES

1. Alexander S. Gillis. (2023). Supervised Learning. Retrieved October 20, 2023, from TechTarget website: https://www.techtarget.com/searchenterpriseai/definition/supervised-learning

2. Kurtis Pykes (2023). Introduction to Unsupervised Learning. Retrieved October 21, 2023 from Datacamp website: https://www.datacamp.com/blog/introduction-to-unsupervised-learning

3. AlindGupta (2023). Semi-Supervised Learning in ML. Retrieved October 20, 2023, from geeksforgeeks website: https://www.geeksforgeeks.org/ml-semi-supervised-learning

4. Anubhav Singh. (2018). Introduction to Reinforcement Learning. Retrieved October 22, 2023, from Datacamp website: https://www.datacamp.com/tutorial/introduction-reinforcement-learning

5. Mrinal Walia. (2022). Overfitting in Machine Learning and Computer Vision. Retrieved October 17, 2023, from the Roboflow website: https://blog.roboflow.com/overfitting-machine-learning-computer-vision/

6. Explainers. (2022). Overfitting in Machine Learning: What It Is and How to Prevent It. Retrieved October 18, 2023, from the Elite Data Science website: https://elitedatascience.com/overfitting-in-machine-learning#how-to-detect

7. dewangNautiyal. (2023). ML | Underfitting and Overfitting. Retrieved October 18, 2023, from geeksforgeeks website: https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

8. irenecasmir. (2023). Regularization by Early Stopping. Retrieved October 19, 2023, from geeksforgeeks website: https://www.geeksforgeeks.org/regularization-by-early-stopping/

9. Jason Brownlee. (2019). Overfitting and Underfitting With Machine Learning Algorithms. Retrieved October 20, 2023, from Machine Learning Mastery website: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/

10. Avijeet Biswal. (2023). The Complete Guide on Overfitting and Underfitting in Machine Learning. Retrieved October 21,2023, from simplilearn website: https://www.simplilearn.com/tutorials/machine-learning-tutorial/overfitting-and-underfitting#what_is_underfitting

11. Benyamin Ghojogh, Mark Crowley. (2023). The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial. Machine Learning Laboratory, University of Waterloo, Waterloo, ON, Canada

12. Steve Lawrence, C. Lee Giles , Ah Chung Tsoi. Lessons in Neural Network Training: Overfitting May be Harder than Expected. NEC Research, Independence Way, Princeton, NJ 08540, Faculty of Informatics, Uni. of Wollongong, Australia

13. Haider Khalaf Jabbar, Dr. Rafiqul Zaman Khan. METHODS TO AVOID OVER-FITTING AND UNDER-FITTING IN SUPERVISED MACHINE LEARNING (COMPARATIVE STUDY). Department of Computer Science Aligarh Muslim University, India

14. Xue Ying. (2019). An Overview of Overfitting and its Solutions. China