UNIVERSIDAD
POLITÉCNICA
DE YUCATÁN

BIS
Universities

Mauricio Daniel Zaldívar Medina

2009147

Machine Learning

Portfolio of Evidence

UNIT 1

Robotics Engineering 9°A

1. Overfitting:

Overfitting occurs when a machine learning model is excessively complex, capturing noise and random fluctuations in the training data rather than the underlying patterns or relationships. In other words, the model fits the training data too closely, to the extent that it memorizes the data points rather than learning the general trends. As a result, an overfitted model performs very well on the training data but poorly on new, unseen data because it has essentially tailored itself to the idiosyncrasies of the training dataset. Overfitting often leads to poor generalization and reduced model performance.

Key characteristics:

- Low training error (the model fits the training data very well).
- High validation or test error (poor performance on new data).
- Complex model with many parameters or features.
- Sensitivity to small changes in the training data.

Solutions:

- Regularization: Use techniques like L1 (Lasso) or L2 (Ridge) regularization to add penalty terms to the model's loss function. This discourages large coefficients and helps prevent overfitting.
- Cross-Validation: Employ cross-validation techniques, such as k-fold cross-validation, to assess model performance on different subsets of the data. This helps identify overfitting by evaluating how well the model generalizes to unseen data.
- Feature Selection: Select a subset of the most relevant features, removing irrelevant or noisy ones. Fewer features can lead to a simpler model less prone to overfitting.
- Reduce Model Complexity: Choose simpler models with fewer parameters, such as linear models or decision trees with limited depth. Decreasing model complexity can mitigate overfitting.

- Increase Data Size: Gathering more data can help the model generalize better, as it has more information to learn from and is less likely to memorize noise.
- Early Stopping: Monitor the model's performance on a validation dataset during training. Stop training when the validation error starts to increase, as this indicates overfitting.

2. Underfitting:

Underfitting, on the other hand, occurs when a machine learning model is too simplistic to capture the underlying patterns in the data. In this case, the model is overly generalized and fails to represent the relationships between the features and the target variable accurately. An underfitted model performs poorly on both the training data and new data because it fails to capture the relevant information and trends present in the data.

Key characteristics:

- High training error (the model does not fit the training data well).
- High validation or test error (poor performance on new data).
- Too simple or insufficiently complex model.
- Inability to capture the underlying patterns or relationships in the data.

Solutions:

- Feature Engineering: Create more informative features or transform existing ones to better represent the data.
- Increase Model Complexity: Use more complex models with additional parameters or layers, like deep neural networks or ensemble methods, to allow the model to capture more intricate relationships in the data.
- Collect More Data: If feasible, gather more data to provide the model with a broader range of examples and patterns to learn from.

- Adjust Hyperparameters: Tune hyperparameters, such as learning rate, batch size, and regularization strength, to find the right balance between model complexity and underfitting.
- Ensemble Methods: Combine multiple simpler models to create a more powerful ensemble model that can capture complex relationships without overfitting.

3.- Outliers:

Outliers are data points that deviate significantly from the majority of data in a dataset. They can have a substantial impact on statistical analyses and machine learning models, often requiring special consideration. Outliers exhibit distinct characteristics that distinguish them from typical data points:

- Extreme Values: Outliers are typically data points that have values that are significantly higher or lower than the values of most other data points in the dataset. These extreme values make them stand out.
- Unusual Observations: Outliers represent observations that are unusual or unexpected based on the patterns and trends exhibited by the rest of the data. They don't conform to the norm.
- Skewed Distribution: The presence of outliers can skew the distribution of data, making it non-normally distributed. This can affect the assumptions underlying many statistical techniques.
- Influence on Summary Statistics: Outliers can significantly affect summary statistics like the mean (average) and standard deviation. The mean is particularly sensitive to extreme values.
- Impact on Visualizations: In graphical representations of data, outliers may appear as data points far removed from the main cluster, making the visualizations harder to interpret.
- Potential Data Errors: Outliers may sometimes indicate data errors, measurement inaccuracies, or data entry mistakes. It's essential to investigate them to determine whether they are genuine or erroneous.

Types of outliers:

- Global Outliers: These are outliers that are outliers across the entire dataset. They deviate significantly from the entire data distribution.
- Contextual Outliers: Contextual outliers are outliers only within a specific context or subgroup of data. In other words, they may not be outliers when considered in isolation but become outliers when compared to a subset of the data.

Solutions:

- Data Cleaning: Identify and remove or correct obvious data errors or outliers that are due to measurement inaccuracies or data entry mistakes.
- Winsorization: Cap or clip extreme values by replacing outliers with the nearest values within a specified range. This helps mitigate the impact of outliers.
- Transformations: Apply data transformations (e.g., logarithmic transformation) to make the data less sensitive to outliers.
- Robust Statistics: Use robust statistical methods that are less influenced by outliers, such as the median instead of the mean or robust regression techniques.
- Model Robustness: Choose machine learning models that are inherently robust to outliers, such as decision trees or random forests.
- Feature Engineering: Create new features or transformations that explicitly account for outliers or their effects on the data.

4.- Dimensionality Problem:

The dimensionality problem, also known as the "curse of dimensionality," is a common challenge encountered in various fields, including statistics, machine learning, data analysis, and optimization. It arises when working with datasets that possess a large number of features or dimensions. In essence, this problem refers to the difficulties and complexities that emerge as the dimensionality of the data increases.

One prominent consequence of high dimensionality is increased computational complexity. As the number of features or dimensions grows, the computational resources required to process and analyze the data also expand significantly. Many algorithms and techniques become computationally intensive or even impractical to apply in such high-dimensional spaces.

Another issue related to high-dimensional data is data sparsity. In these spaces, data points tend to become sparser, meaning that there are fewer actual data points compared to the vast number of possible data points. This sparsity can lead to problems like overfitting, where models struggle to generalize from limited data, resulting in suboptimal performance. Moreover, the dimensionality problem necessitates larger sample sizes to maintain statistical validity. With more dimensions, a greater amount of data is needed to obtain reliable estimates of relationships between variables. This requirement for larger datasets can be a practical challenge in many real-world scenarios.

In high-dimensional spaces, visualization becomes problematic. While it is relatively easy to visualize data in two or three dimensions, as the dimensionality increases, it becomes increasingly challenging or even impossible to create effective visual representations. This limitation makes it difficult to gain insights from graphical analysis, a common tool for understanding data. Additionally, the curse of dimensionality is closely associated with the risk of overfitting. In high-dimensional spaces, models can easily fit the noise in the data rather than capturing the true underlying patterns. As a result, these models often perform poorly when applied to new, unseen data, undermining their generalization capabilities.

To address the dimensionality problem, several strategies and techniques can be employed. These include feature selection, which involves choosing a subset of the most relevant features while discarding less important ones. Feature engineering can also be applied to create new features or transformations that reduce dimensionality in a more meaningful way. Dimensionality reduction techniques like Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) aim to preserve as much information as possible while reducing dimensionality. Regularization methods, such as L1 regularization (Lasso), can automatically select important features and reduce dimensionality. Furthermore, domain knowledge can guide feature selection and engineering, ensuring a focus on the most relevant variables for the specific problem at hand. Data preprocessing techniques, such as feature scaling or normalization, can help algorithms perform better in high-dimensional spaces.

5.- Dimensionality reduction:

Dimensionality reduction is a crucial technique in data analysis and machine learning that aims to reduce the number of features or dimensions in a dataset while preserving as much relevant information as possible. It becomes particularly valuable when working with high-dimensional data, where a large number of features can lead to computational challenges, increased risk of overfitting, and difficulties in visualization. The dimensionality reduction process typically involves several key steps.

First, the process begins with data collection and preprocessing. You start with a dataset that contains numerous features, and it's essential to ensure that the data is cleaned and prepared for analysis. This includes handling missing values, scaling or normalizing features, and addressing outliers if they exist. Before applying dimensionality reduction techniques, you may consider feature selection. This step involves identifying and retaining only the most relevant features, effectively reducing dimensionality without resorting to more advanced techniques. Next, you choose an appropriate dimensionality reduction method based on your specific dataset and goals. Several common techniques are available, including Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), Linear Discriminant Analysis (LDA), and autoencoders for deep learning-based reduction.

The chosen dimensionality reduction technique is then applied to your dataset. For example, PCA involves calculating the principal components and projecting the data onto a reduced-dimensional subspace, while t-SNE focuses on preserving pairwise similarities between data points in the reduced space. LDA is used in supervised settings to maximize the separability of different classes, and autoencoders employ neural networks to learn a lower-dimensional representation of the data.

When using PCA or similar methods that require specifying the number of dimensions, you need to determine the appropriate number of components to retain. This decision can be based on various factors, such as explained variance, scree plots, or domain knowledge. After the dimensionality reduction is complete, you should evaluate the quality and effectiveness of the reduced data. Assess how much variance or information is retained compared to the original data, and examine how well the reduced data represents the underlying patterns or relationships in your dataset.

Once you are satisfied with the dimensionality reduction results, you can use the reduced data for various purposes, such as machine learning, visualization, or further analysis. It's important to monitor for overfitting, especially if dimensionality reduction is part of a machine learning pipeline. While reducing dimensionality can help combat overfitting, it's not a guaranteed solution, and other techniques like regularization and cross-validation should still be applied to ensure model generalization.

Lastly, documenting the dimensionality reduction process, including the chosen method, the number of dimensions/components retained, and any parameters or settings used, is essential for reproducibility and transparency in your data analysis or modeling projects.

6.-Bias-variance Trade-Off

The bias-variance trade-off is a central concept in the field of machine learning and statistical modeling, playing a crucial role in determining the performance of predictive models. At its core, this trade-off represents the balance between two types of errors that affect the model's ability to generalize to unseen data: bias and variance.

Bias, in the context of this trade-off, signifies the error introduced when a model simplifies a real-world problem by making strong assumptions or approximations. High-bias models are characterized by their simplicity and their limited capacity to capture complex relationships within the data. These models often underfit the training data, resulting in high training error. They tend to generalize poorly because they overlook important patterns and nuances in the data.

On the other side of the trade-off, we have variance, which represents the error stemming from a model's sensitivity to minor fluctuations or noise in the training data. High-variance models are flexible and capable of fitting the training data closely, including the noise and random variations present in the dataset. These models, however, can overfit the data, meaning they capture the noise rather than the underlying patterns, leading to high test error when applied to new, unseen data.

Achieving the right balance between bias and variance is essential for developing models that generalize effectively. High-bias models are relatively simple and make strong assumptions about the data, resulting in low variance but often leading to poor performance due to underfitting. Conversely, high-variance models are complex and capable of fitting the training data very closely, but they tend to overfit and exhibit poor generalization. The challenge is to find a middle ground that minimizes both bias and variance.

Several strategies can be employed to manage the bias-variance trade-off effectively. Model selection is one key aspect, involving the choice of an appropriate model with an optimal level of complexity for the given problem. Hyperparameter tuning is another critical step, where adjustments to parameters like regularization strength or learning rate can help control the trade-off between bias and variance. Cross-validation techniques, such as k-fold cross-validation, assist in estimating a model's performance on unseen data and guide decisions regarding model complexity.

Ensemble methods are also valuable tools for addressing the bias-variance trade-off. By combining multiple models, ensemble methods can reduce variance while preserving the strengths of individual models, ultimately improving generalization.

Conclusion:

In conclusion, the concepts of overfitting, underfitting, outliers, the dimensionality problem, and the bias-variance trade-off are integral to the field of data analysis and machine learning. These concepts collectively underscore the complexity and intricacy of working with data, highlighting the delicate balance required in model development and data preprocessing. They serve as guiding principles for practitioners, emphasizing the need for careful consideration, domain expertise, and nuanced decision-making throughout the data analysis and modeling process.

In this dynamic and evolving landscape, these challenges are not obstacles but opportunities. They challenge us to strike the right balance between complexity and simplicity, outliers and insights, and bias and variance. Embracing these concepts fosters a deeper understanding of data's potential and paves the way for more accurate predictions, meaningful insights, and informed decision-making in the ever-expanding world of data science and machine learning.

REFERENCES:

- *Overfitting and Underfitting With Machine Learning Algorithms - MachineLearningMastery.com*. (s.f.). MachineLearningMastery.com. https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/
- *Overfitting and underfitting in machine learning | SuperAnnotate*. (s.f.). The ultimate training data platform for AI | SuperAnnotate. https://www.superannotate.com/blog/overfitting-and-underfitting-in-machine-learning
- Nikolaiev, D. (2022, 2 de noviembre). *Overfitting and Underfitting Principles*. Medium. https://towardsdatascience.com/overfitting-and-underfitting-principles-ea8964d9c45c
- *ML | Underfitting and Overfitting - GeeksforGeeks*. (s.f.). GeeksforGeeks. https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/
- Lemonaki, D. (2021, 24 de agosto). *What is an Outlier? Definition and How to Find Outliers in Statistics*. freeCodeCamp.org. https://www.freecodecamp.org/news/what-is-an-outlier-definition-and-how-to-find-outliers-in-statistics/
- Khaciyants, I. L. A. (2023, 13 de marzo). *What Is the Bias-Variance Tradeoff in Machine Learning?* Serokell Software Development Company. https://serokell.io/blog/bias-variance-tradeoff
- *Introduction to Dimensionality Reduction Technique - Javatpoint*. (s.f.). www.javatpoint.com. https://www.javatpoint.com/dimensionality-reduction-technique
- *Bias-Variance Trade Off - Machine Learning - GeeksforGeeks*. (s.f.). GeeksforGeeks. https://www.geeksforgeeks.org/ml-bias-variance-trade-off/
- *7.1.6. What are outliers in the data?* (s.f.). Information Technology Laboratory | NIST. https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm