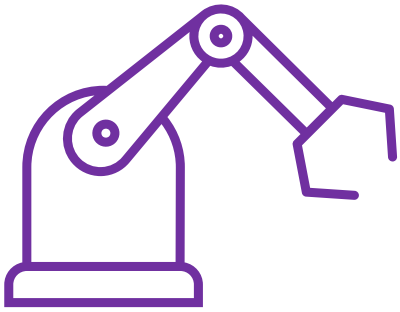
Solution to most common problems in ML

Portfolio evidence



Teacher: Victor Ortiz

Name: Eduardo Antonio Flores Arellano

N. Control: 2009055

Group: 9° A Robotics

University:

Polytechnical University of Yucatán

Overfitting

Overfitting is an undesirable behavior often observed in machine learning, wherein a model excels at providing precise predictions for its training data but falters when tasked with making predictions on fresh, previously unseen data. Data scientists typically initiate the machine learning journey by training the model using a well-established dataset. Subsequently, leveraging the insights gained during this training, the model endeavors to anticipate outcomes for entirely new datasets. An overfit model, however, tends to yield inaccurate predictions and struggles to adapt to various types of new data.

- **Cross-Validation:** Implement k-fold cross-validation to evaluate the model's performance across multiple data subsets, ensuring it generalizes effectively.
- **Regularization:** Apply techniques like L1 (Lasso) or L2 (Ridge) regularization to penalize complex model parameters and mitigate overfitting.
- **Feature Selection:** Carefully choose relevant features while eliminating unnecessary ones to reduce model complexity.
- **Increase Data:** Augment the dataset with more diverse and representative data points to facilitate generalized pattern learning.

Underfitting

Underfitting represents another category of errors encountered in machine learning when the model fails to establish a substantial connection between the input and output data. Models tend to underfit when they haven't undergone adequate training for a sufficient duration on a substantial volume of data points.

- **Model Complexity:** opt for more complex models capable of capturing intricate data relationships.
- **Feature Engineering:** Create new features or transform existing ones to provide the model with richer information.
- **Data Augmentation:** Expand the dataset through techniques like data synthesis or oversampling.
- **Hyperparameter Tuning:** Adjust model hyperparameters, such as learning rates and depths, for fine-tuning model performance.

Outliers

In the machine learning pipeline, data cleaning and preprocessing are crucial steps for gaining a better understanding of the data. During this phase, you address missing values, detect outliers, and more.

Outliers, with their extreme values, can significantly skew statistical analyses on the dataset, potentially leading to less effective models. Dealing with outliers requires domain expertise and a clear understanding of the data distribution and use case.

When data or specific dataset features conform to a normal distribution, identifying outliers can be accomplished by employing the data's standard deviation or the corresponding z-score.

In statistical terms, the standard deviation quantifies the dispersion of data relative to the mean, essentially gauging the extent to which data points deviate from the mean.

In the case of normally distributed data, approximately 68.2% of the data falls within one standard deviation from the mean, while roughly 95.4% and 99.7% lie within two and three standard deviations from the mean, respectively.

- **Data Preprocessing:** Utilize techniques like scaling, normalization, or winsorization to mitigate the impact of outliers.
- **Outlier Detection:** Employ algorithms such as Z-score, IQR, or machine learning-based methods to identify and handle outliers.
- **Data Truncation:** Consider capping extreme values or transforming outliers to bring them within a reasonable range.
- **Robust Models**: Implement robust models like decision trees or random forests, which are less sensitive to outliers.

The dimensionality problem

The dimensionality problem, often encountered in various fields such as machine learning, statistics, and data analysis, refers to the challenges and issues that arise when working with datasets that have many features or variables. In essence, it is the problem of dealing with high-dimensional data.

Dimensionality reduction is a crucial data preprocessing technique used to address the dimensionality problem by reducing the number of features or variables in a dataset while preserving the essential information and patterns within the data. This process simplifies data analysis, visualization, and modeling, making it more manageable and efficient. Here's an overview of the dimensionality reduction process:

Data Preparation:

- Begin with a dataset that has a high number of features or dimensions.
- Ensure the data is cleaned and preprocessed to handle missing values, outliers, and other data quality issues.

Feature Selection or Extraction:

Feature Selection: This approach involves selecting a subset of the original features while discarding the less relevant or redundant ones. Common techniques for feature selection include:

- Univariate feature selection: Selecting features based on statistical tests or scores.
- Recursive Feature Elimination (RFE): Iteratively removing the least important features.
- Feature importance from tree-based models: Identifying important features using decision trees or random forests.

Feature Extraction: In this approach, new features are generated by transforming the original features into a lower-dimensional space. Common techniques for feature extraction include:

- Principal Component Analysis (PCA): A linear technique that identifies orthogonal axes (principal components) along which the data varies the most, effectively reducing dimensionality.
- Linear Discriminant Analysis (LDA): Maximizes the separability between different classes in supervised classification problems.
- t-Distributed Stochastic Neighbor Embedding (t-SNE): Used for nonlinear dimensionality reduction and visualization of high-dimensional data.
- Autoencoders: Neural network-based models that learn a compressed representation of the data through encoding and decoding layers.

Evaluation:

- It's essential to evaluate the effectiveness of dimensionality reduction in preserving the relevant information while reducing noise.
- Evaluation can involve comparing the performance of machine learning models before and after dimensionality reduction or assessing the variance retained in the case of feature extraction techniques like PCA.

Application:

- Once dimensionality reduction is applied and validated, the reduced-dimension dataset can be used for various purposes, such as data analysis, visualization, or building predictive models.
- Reduced-dimensional data is typically more manageable and can lead to improved model generalization and interpretability.

Monitoring and Maintenance:

- Continuously monitor the impact of dimensionality reduction on your analysis or models.
- If necessary, update the dimensionality reduction approach or parameters to adapt to changing data characteristics or research goals.

Bias-variance trade-off

The bias-variance trade-off is a fundamental concept in machine learning. It refers to the trade-off or balance that must be struck between two types of errors that a predictive model can make when fitting to data:

Bias: A model with high bias makes simplifying assumptions about the data and tends to underestimate the complexity of the problem. This can result in poor fit to the training data and an inability to capture important patterns, known as "underfitting."

Variance: On the other hand, a model with high variance is highly sensitive to fluctuations in the training data. This can lead the model to overreact to these fluctuations, including noise in the data, and not generalize well to new data, termed "overfitting."

The challenge is to find the right balance between these two extremes. This means choosing the model's complexity in a way that captures the underlying structure in the data without overfitting. Some strategies to address this trade-off include:

Cross-Validation: Using techniques like cross-validation to assess how the model performs on unseen data and adjusting its complexity accordingly.

Regularization: Applying regularization techniques such as L1 (Lasso) or L2 (Ridge) that penalize overly complex models, reducing variance.

Feature Selection: Carefully choosing relevant features and eliminating irrelevant ones to simplify the model and reduce bias.

Ensemble Approaches: Using ensemble methods like random forests or gradient boosting to reduce variance by averaging multiple models.

References

1. Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python. O'Reilly Media.
2. What is overfitting? - Overfitting in machine learning explained - AWS. (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/what-is/overfitting/#:~:text=Overfitting%20is%20an%20undesirable%20machine,but%20not%20for%20new%20data.
3. C, B. P. (2022). How to Detect Outliers in Machine Learning – 4 methods for Outlier Detection. freeCodeCamp.org. https://www.freecodecamp.org/news/how-to-detect-outliers-in-machine-learning/
4. Zheng, A., & Casari, A. (2018). Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. O'Reilly Media.