2023

Solution to most common problems in ML - Portfolio evidence

JESUS ISRAEL PRADO PINEDA

Universidad Politécnica de Yucatán

15-9-2023

Define the concepts of: Overfitting & Underfitting.

Overfitting and underfitting are two important concepts in machine learning and statistics that describe the performance of a predictive model in relation to the data it is trained on:

1. Overfitting:

Overfitting occurs when a machine learning model learns the training data too well, capturing not only the underlying patterns but also noise and random fluctuations present in the data. As a result, an overfit model performs exceptionally well on the training data but poorly on unseen or new data. It essentially memorizes the training data rather than generalizing from it.

Characteristics of overfitting:

- Low training error (model fits the training data very closely).
- High test or validation error (poor performance on new data).
- The model is overly complex, with too many parameters.
- The model may exhibit wild and erratic predictions when applied to new data.
- It can be thought of as a form of "overlearning."

Overfitting can be mitigated by techniques such as using simpler models, adding more training data, applying regularization methods, or using cross-validation to evaluate model performance.

2. Underfitting:

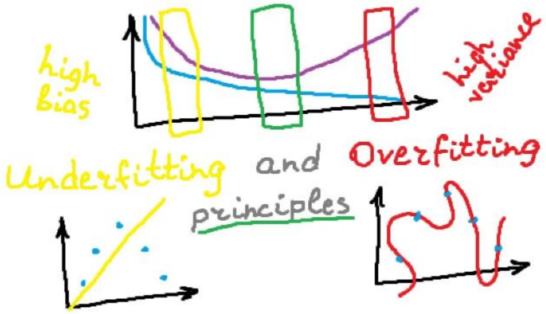
Underfitting, on the other hand, occurs when a machine learning model is too simplistic to capture the underlying patterns in the data. An underfit model performs poorly not only on the training data but also on new data because it fails to capture the complexity of the underlying relationships in the data.

Characteristics of underfitting:

- High training error (model does not fit the training data well).
- High test or validation error (poor performance on new data).
- The model is too simple or has too few parameters to represent the data adequately.
- It may fail to capture important features or patterns in the data.

Underfitting can be addressed by using more complex models, increasing model capacity, or selecting more relevant features from the data. Additionally, increasing the amount of training data can help in some cases.

The goal in machine learning is to strike a balance between overfitting and underfitting by building models that generalize well to new, unseen data. This balance is crucial for creating models that provide accurate and reliable predictions. Techniques like hyperparameter tuning and cross-validation play a significant role in achieving this balance.



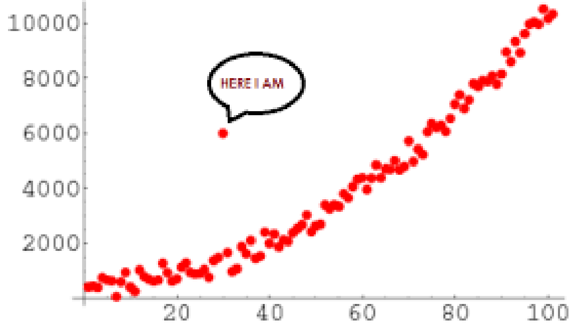
Define and distinguish the characteristics of outliers.

An *outlier* is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Before abnormal observations can be singled out, it is necessary to characterize normal observations.

Two activities are essential for characterizing a set of data:

A) Examination of the overall shape of the graphed data for important features, including symmetry and departures from assumptions. The chapter on Exploratory Data Analysis (EDA) discusses assumptions and summarization of data in detail.

B) Examination of the data for unusual observations that are far removed from the mass of data. These points are often referred to as outliers. Two graphical techniques for identifying outliers, scatter plots and box plots, along with an analytic procedure for detecting outliers when the distribution is normal (Grubbs' Test), are also discussed in detail in the EDA chapter.



Discuss the most common solutions for overfitting, underfitting and presence of outliers in datasets.

A) Overfitting:

Use Simpler Models*:* One of the most effective ways to combat overfitting is to use simpler models with fewer parameters. For example, if you're using a deep neural network, you can reduce the number of layers or neurons.

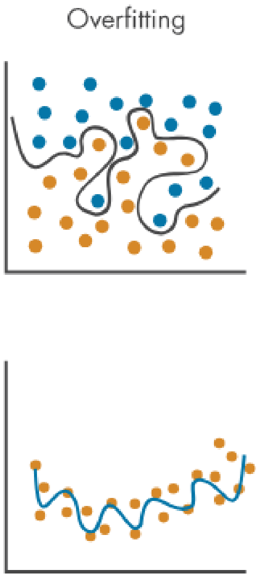
Regularization: Regularization techniques like L1 and L2 regularization (also known as Lasso and Ridge regression, respectively) add penalty terms to the model's loss function, discouraging it from assigning excessively high weights to features. This helps prevent overfitting.

Cross-Validation*:* Employ cross-validation techniques such as k-fold cross-validation to assess how well your model generalizes to unseen data. This can help you identify and mitigate overfitting.

Feature Selection*:* Carefully select and engineer features, discarding irrelevant or redundant ones. Feature selection can simplify the model and reduce the risk of overfitting.

Early Stopping: Monitor the model's performance on a validation set during training. Stop training when the performance on the validation set starts to degrade, indicating that the model is starting to overfit.

Ensemble Methods: Use ensemble methods like bagging (e.g., Random Forests) or boosting (e.g., Gradient Boosting) to combine multiple models and reduce overfitting.



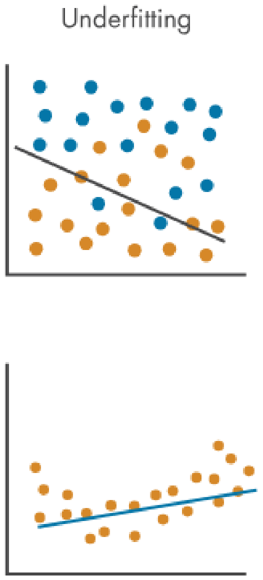
B) Underfitting:

Increase Model Complexity*:* If your model is underfitting, it may be too simple to capture the underlying patterns in the data. Try increasing the model's complexity by adding more layers, neurons, or using a more sophisticated algorithm.

Feature Engineering: Consider creating new features that better represent the relationships in the data. Sometimes, underfitting occurs because essential features are missing.

Collect More Data: If possible, gather more data to provide the model with a richer source of information. Additional data can help the model better understand the underlying patterns.

Adjust Hyperparameters: Experiment with different hyperparameter settings, such as learning rate, batch size, or the number of epochs, to fine-tune the model's performance.



C) Presence of Outliers:

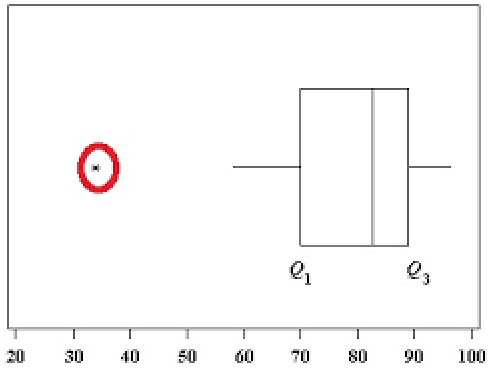
Outlier Detection: Use statistical methods or machine learning algorithms to identify and flag potential outliers in your dataset. Common approaches include the Z-score, modified Z-score, and clustering-based methods like DBSCAN.

Data Transformation: Consider transforming the data to make it more robust to outliers. For example, you can apply log transformations to skewed data or use robust scalers like the Robust-Scaler in preprocessing.

Winsorization: Replace extreme outliers with less extreme values, such as the 5th and 95th percentiles, to reduce the impact of outliers on your model.

Outlier Handling: Depending on the context, you may choose to remove outliers if they are likely to be errors or keep them if they represent genuine data points. The decision should be based on domain knowledge and the specific problem you're addressing.

Robust Models: Consider using machine learning models that are less sensitive to outliers, such as robust regression techniques or ensemble methods that can down weight the influence of outliers.



Describe the dimensionality problem.

The dimensionality problem, often referred to as the curse of dimensionality, is a phenomenon in data analysis, machine learning, and statistics that arises when dealing with high-dimensional data. It refers to the challenges and issues that occur when working with datasets that have a large number of features or dimensions. The dimensionality problem can lead to several difficulties and complexities, including:

A) Increased Computational Complexity: As the number of dimensions in the dataset increases, computational requirements grow exponentially. Many algorithms and techniques that work well in low-dimensional spaces become computationally infeasible or very slow in high-dimensional spaces. This can significantly impact model training times and overall efficiency.

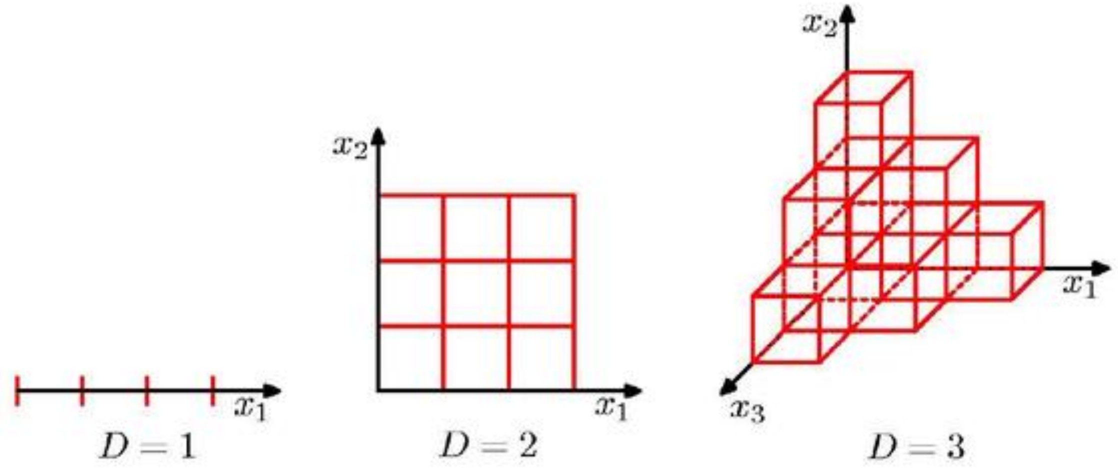
B) Data Sparsity: High-dimensional spaces are often sparsely populated, meaning that data points become increasingly distant from each other as dimensions increase. This sparsity can make it difficult to find meaningful patterns or relationships in the data, as there may not be enough data points in the relevant regions of the space.

C) Overfitting: In high-dimensional spaces, models are more prone to overfitting. With many features, a model can easily find spurious correlations in the training data that do not generalize well to new, unseen data. Overfit models can be overly complex and perform poorly on validation or test data.

D) Increased Data Requirements: To achieve reliable statistical significance and avoid overfitting, high-dimensional datasets often require a much larger number of samples. This means that collecting sufficient data can be more challenging and expensive.

E) Curse of Dimensionality in Distance Metrics: Distance-based similarity measures (e.g., Euclidean distance) become less meaningful in high-dimensional spaces. In such spaces, all data points appear to be roughly equidistant from each other, which can make clustering and nearest-neighbor searches less effective.

F) Feature Engineering Challenges: In high-dimensional data, feature selection and feature engineering become more critical and complex. Identifying which features are relevant and informative can be challenging, and dimensionality reduction techniques may be needed to simplify the dataset.



Describe the dimensionality reduction process.

Dimensionality reduction is a process in data analysis and machine learning that involves reducing the number of features or dimensions in a dataset while preserving as much relevant information as possible. It is commonly used to address the curse of dimensionality, improve model efficiency, and enhance the interpretability of data. The dimensionality reduction process typically involves the following steps:

A) Data Preparation:

Start with a dataset that contains a large number of features or dimensions. This could be a dataset with a high-dimensional feature space, such as text data with many words or documents with many attributes.

Perform any necessary data preprocessing, including handling missing values, scaling or normalizing features, and encoding categorical variables.

B) Feature Selection (Optional):

Before applying dimensionality reduction techniques, consider performing feature selection to eliminate irrelevant or redundant features. Feature selection helps reduce dimensionality without altering the original feature space significantly.

Feature selection methods can be filter-based (e.g., correlation-based feature selection) or wrapper-based (e.g., recursive feature elimination).

C) Choose a Dimensionality Reduction Technique:

Select an appropriate dimensionality reduction technique based on the characteristics of your data and the goals of your analysis. Two common methods are Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE), but there are others like Linear Discriminant Analysis (LDA) and autoencoders.

D) Apply Dimensionality Reduction:

Implement the chosen dimensionality reduction technique to transform the original high-dimensional data into a lower-dimensional representation.

For PCA, this involves finding the principal components (linear combinations of the original features) that capture the most variance in the data. The number of principal components to retain is a user-defined parameter.

For t-SNE, it focuses on preserving pairwise similarities between data points in the lower-dimensional space.

E) Determine the Reduced Dimensionality:

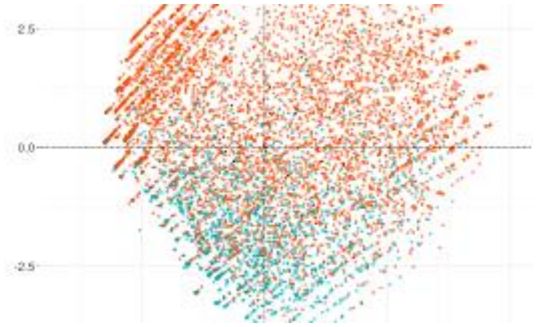
Specify the desired reduced dimensionality or the number of components (e.g., principal components or t-SNE dimensions) to retain. This can be based on the desired level of information preservation or computational constraints.

F) Transform Data:

Apply the dimensionality reduction transformation to the entire dataset to obtain the reduced-dimensional representation of the data.

G) Evaluate Information Retention:

Assess the extent to which the reduced-dimensional representation preserves the essential information and patterns from the original data. This can be done using various metrics, such as explained variance (for PCA) or visualization techniques.



Explain the bias-variance trade-off.

The bias-variance trade-off is a fundamental concept in machine learning and statistics that helps us understand the balance between two types of errors that a predictive model can make: bias and variance. It relates to the model's ability to generalize from training data to unseen data, Bias refers to the error introduced by approximating a real-world problem (which may be complex) by a simplified model. A model with high bias makes strong assumptions about the underlying data distribution, resulting in systematic errors. In other words, it underfits the data. High bias can lead to poor predictive performance because the model is too simple to capture the true underlying patterns in the data.

Characteristics of a high-bias model:

- It performs poorly on both the training data and unseen data.

- It oversimplifies the problem and misses important relationships in the data.

- It has a high training error.

Variance: Variance, on the other hand, refers to the error introduced by the model's sensitivity to small fluctuations or noise in the training data. A model with high variance is highly flexible and can capture intricate details in the training data, but it may not generalize well to new, unseen data. High variance can lead to overfitting.

Characteristics of a high-variance model:

- It performs very well on the training data but poorly on unseen data.

- It captures noise in the data rather than the true underlying patterns.

- It has a low training error but a high-test error.

The trade-off between bias and variance can be summarized as follows:

Low Complexity Models (High Bias, Low Variance): Simpler models with fewer parameters tend to have higher bias but lower variance. They make strong assumptions and generalize poorly to complex data.

High Complexity Models (Low Bias, High Variance): Complex models with many parameters tend to have lower bias but higher variance. They can fit the training data very closely but may not generalize well to new data.

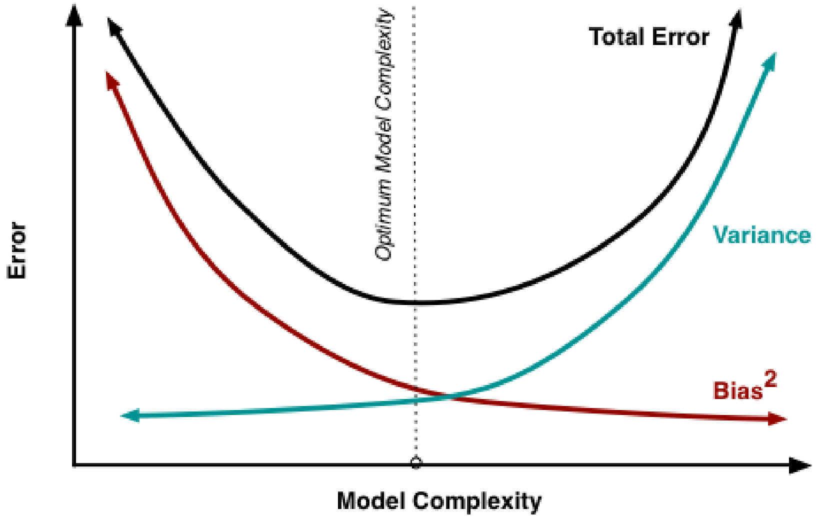
The goal in machine learning is to strike a balance between bias and variance, finding the optimal level of model complexity that minimizes the overall prediction error on unseen data. This is often achieved through techniques such as:

* Regularization: Introducing penalties for complex models to discourage overfitting and reduce variance (e.g., L1 and L2 regularization).

- Cross-Validation: Using techniques like k-fold cross-validation to estimate how well a model generalizes to new data and selecting the model that performs best on validation data.

- Ensemble Methods: Combining multiple models (e.g., Random Forests, Gradient Boosting) to reduce variance by averaging or combining their predictions.

Understanding the bias-variance trade-off is crucial for model selection and hyperparameter tuning, as it helps in choosing the right level of model complexity to achieve good generalization performance.



References

7.1.6. What are outliers in the data? (s. f.). https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm

Face recognition using extended ISOMaP. (2002). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/1039901

GeeksforGeeks. (2023). Introduction to dimensionality reduction. *GeeksforGeeks*. https://www.geeksforgeeks.org/dimensionality-reduction/#:~:text=Important%20points%3A-,Dimensionality%20reduction%20is%20the%20process%20of%20reducing%20the%20number%20of,easier%20to%20visualize%20the%20data.

Minhas, M. S. (2022, 6 enero). Techniques for handling underfitting and overfitting in machine learning. *Medium*. https://towardsdatascience.com/techniques-for-handling-underfitting-and-overfitting-in-machine-learning-348daa2380b9#:~:text=In%20this%20situation%2C%20the%20best,the%20patterns%20in%20the%20data.

Nikolaiev, D. (2022, 8 julio). Overfitting and underfitting principles | by DiMid | towards data science. *Medium*. https://towardsdatascience.com/overfitting-and-underfitting-principles-ea8964d9c45c#:~:text=Underfitting%20means%20that%20your%20model,val%2Ftest%20error%20is%20large.

Singh, S. (2023, 28 junio). Understanding the Bias-Variance tradeoff - towards data science. *Medium*. https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229