MACHINE LEARNING

Shoaib Farooq

Department of Computer Science

DATA INSIGHT
Let us unfold power of data

Instructor Information

Instructor	Shoaib Farooq	E-mail	m.shoaib1050@gmail.com
 GitHub	https://github.com/SHOAIB1050		
 YouTube	https://www.youtube.com/c/pakithub		
 LinkedIn	https://www.linkedin.com/in/shoaib-farooq-b5b190105/		

Let's Start .....

Lecture #10

GOALS

This Lecture Will Cover:

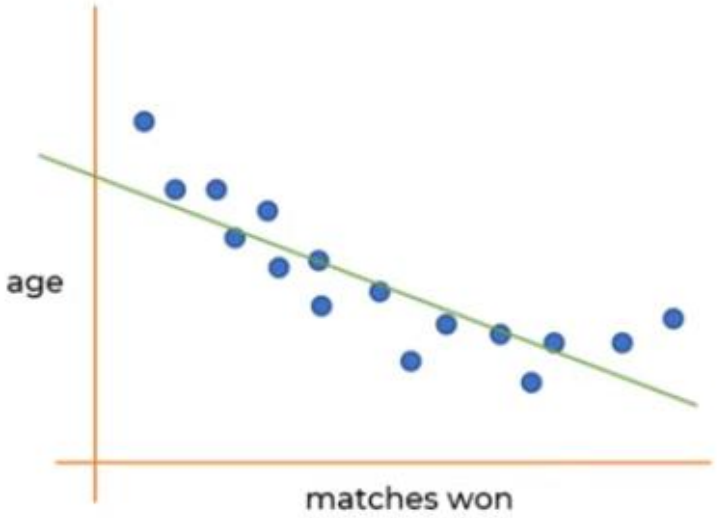
- **Regularization**
- **Balanced and Imbalanced Datasets**

REGULARIZATION

Regularization is a technique used in machine learning to prevent overfitting, which occurs when a model learns the noise in the training data rather than the underlying pattern. When a model is too complex or has too many parameters, it might fit the training data very well but fail to generalize to new, unseen data. Regularization helps by adding a penalty term to the loss function, discouraging the model from fitting the noise and ensuring better generalization.

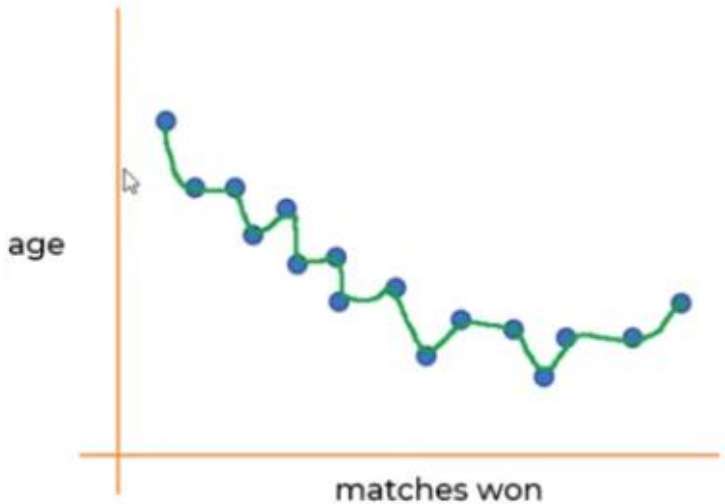
REGULARIZATION

underfit



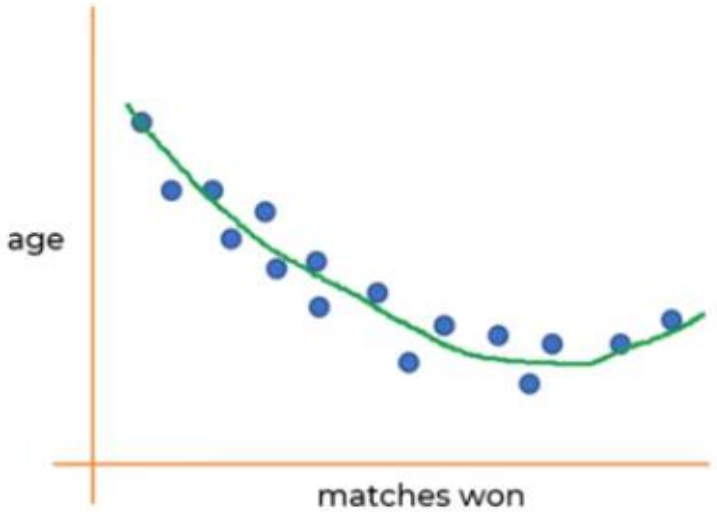
$$\text{match won} = \theta_0 + \theta_1 * age$$

overfit



$$\text{match won} = \theta_0 + \theta_1 * age \quad +\theta_2 * age^2 \quad +\theta_3 * age^3 \quad +\theta_4 * age^4$$

balanced fit

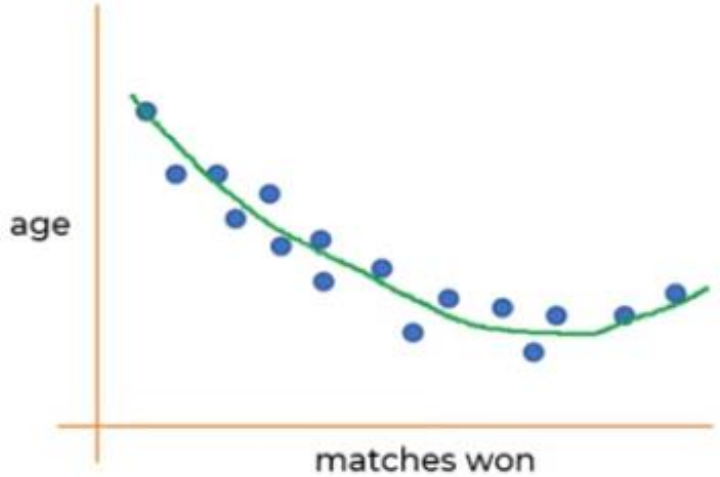


$$\text{match won} = \theta_0 + \theta_1 * age \quad +\theta_2 * age^2$$

HOW REDUCE OVERFITTING

$$\text{match won} = \theta_0 + \theta_1 * age \quad + \theta_2 * age^2 + \theta_3 * age^3 + \theta_4 * age^4$$

Try to make θ_3 and θ_4 almost close to zero



$$\text{match won} = \theta_0 + \theta_1 * age \quad + \theta_2 * age^2$$

L1 REGULARIZATION

$$ms\varepsilon = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x_i)\right)^2 + \lambda \sum_{i=1}^{n} |\theta_i|$$

$$h_\theta(x_i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$$

L1 REGULARIZATION (LASSO REGRESSION)

Explanation: Adds a penalty term proportional to the absolute value of the coefficients.

Loss Function:

$$\text{Loss} = \text{MSE} + \lambda \sum |w_i|$$

λ is a regularization parameter controlling the strength of the penalty.
Effect: Encourages sparsity (i.e., some feature weights become zero), which can be useful for feature selection.

L2 REGULARIZATION (RIDGE REGRESSION)

Explanation: Adds a penalty term proportional to the square of the coefficients.
Loss Function:

$$\text{Loss} = \text{MSE} + \lambda \sum w_i^2$$

Effect: Prevents large weights, leading to simpler models with smaller coefficients. L2 regularization does not produce sparse models but instead shrinks all coefficients evenly.

L2 REGULARIZATION

$$ms\varepsilon = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x_i)\right)^2 + \lambda \sum_{i=1}^{n} \theta_i^2$$

$$h_\theta(x_i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3$$

ELASTIC NET REGULARIZATION

Explanation: Combines both L1 and L2 regularization.
Loss Function:

$$\text{Loss} = \text{MSE} + \lambda_1 \sum |w_i| + \lambda_2 \sum w_i^2$$

Effect: Useful when there are many correlated features, balancing between L1's sparsity and L2's stability

HOW REGULARIZATION WORKS

- Regularization adds a penalty to the model's loss function, discouraging it from fitting the data too closely.
- The parameter λ (or alpha) controls the strength of the regularization. A higher λ means more regularization, resulting in smaller weights and simpler models.
- Choosing an appropriate λ is crucial and is often done through techniques like cross-validation.

OTHER REGULARIZATION TECHNIQUES

1. Drops out Layer

- Randomly "drops out" a fraction of neurons during training, reducing their reliance on any specific feature.
- Prevents overfitting in neural networks by ensuring the model does not rely on specific neurons too heavily

OTHER REGULARIZATION TECHNIQUES

2. Early Stopping

- Stops training when the performance on a validation set starts to degrade.
- Prevents the model from learning noise in the later epochs of training.

3. Batch Normalization

- Normalizes the inputs of each layer in a neural network to have a mean of zero and standard deviation of one.
- Acts as a form of regularization by reducing the model's sensitivity to weight changes.

BALANCED AND IMBALANCED DATASETS

- **Balanced Dataset**

A dataset is considered **balanced** when:

- ✓ The classes are distributed **approximately equally**.
- ✓ For example, if you have a binary classification problem (say, spam vs. not spam emails), a balanced dataset would have nearly the same number of spam and non-spam examples.

- **What is an Imbalanced Dataset?**

A dataset is **imbalanced** when:

- ✓ One or more classes are **underrepresented** compared to others.
- ✓ For instance, in a dataset for **fraud detection**, the number of fraudulent transactions might be much smaller (e.g., 1%) than the number of legitimate transactions (99%).

EVALUATION METRICS FOR IMBALANCED DATA

When dealing with imbalanced datasets, metrics like **accuracy** are often not very informative. Here are better metrics to evaluate performance

- ✓ **Precision**
- ✓ **Recall (Sensitivity or True Positive Rate)**
- ✓ **F1 Score**
- ✓ **ROC-AUC (Receiver Operating Characteristic - Area Under Curve)**

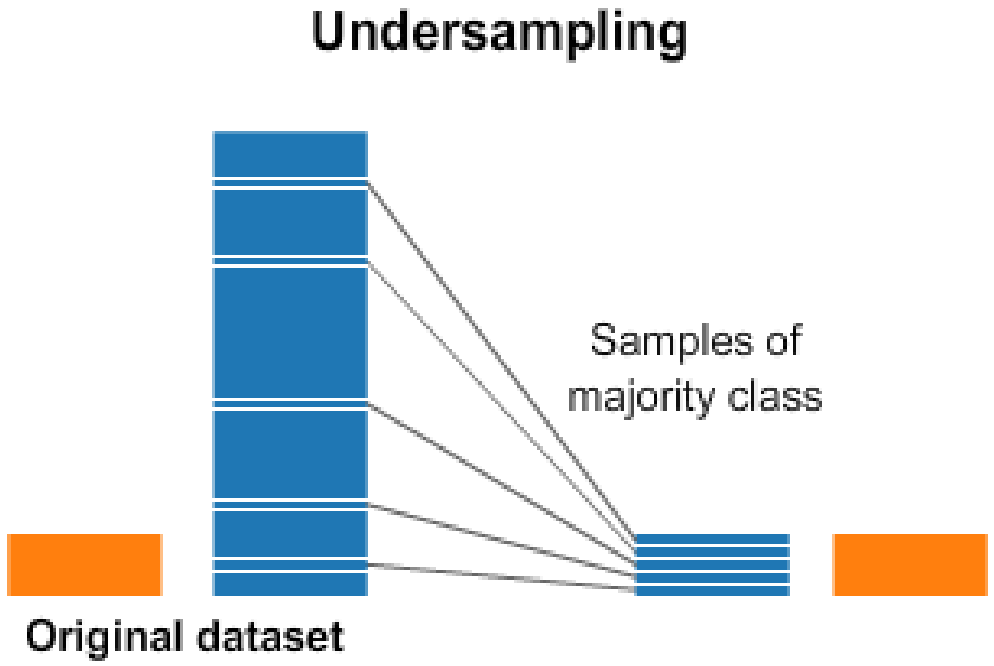
Measures the model's ability to distinguish between classes across various threshold settings.

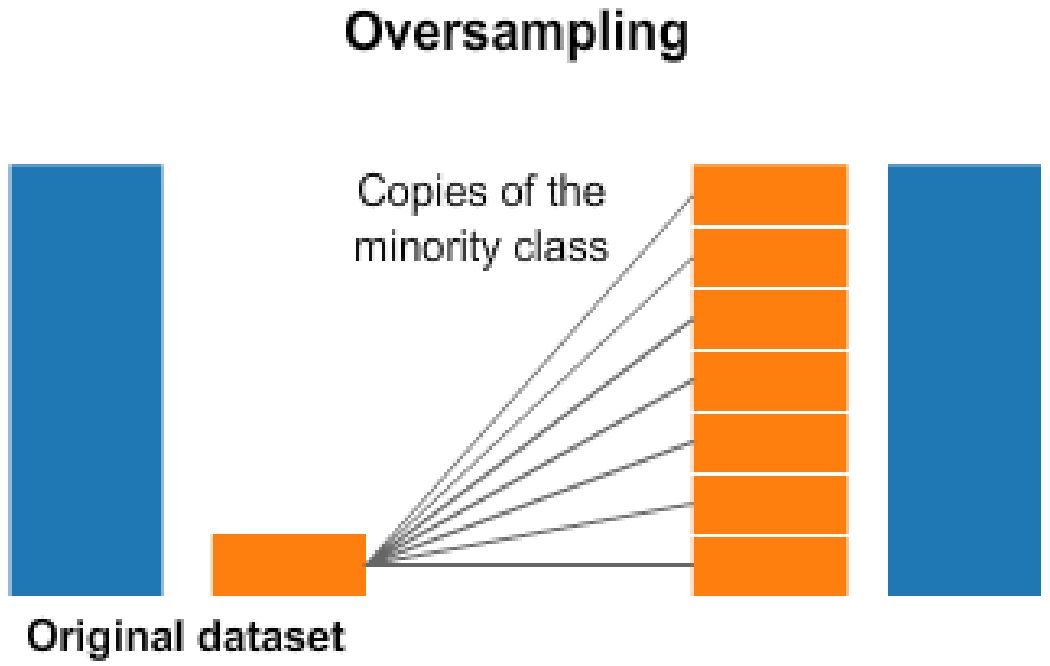
- ✓ **Confusion Matrix:**

Provides a summary of prediction results, breaking down the number of true positives, true negatives, false positives, and false negatives.

BALANCED AND IMBALANCED DATASETS

Balancing imbalanced datasets is crucial to ensure that machine learning models can effectively learn patterns from both the minority and majority classes. Two common strategies to handle imbalanced data are **under sampling** and **oversampling**.





UNDERSAMPLING

Undersampling is a technique used to balance class distribution by **reducing the number of examples in the majority class**. The goal is to make the dataset more balanced without altering the distribution of the minority class.

- **Common Techniques for Undersampling**
- ✓ **Random Undersampling**: Randomly removes samples from the majority class.
- ✓ **Cluster Centroids**: Uses clustering techniques (e.g., k-means) to replace a group of majority class samples with their centroid, reducing data while retaining structure.

ADVANTAGES OF UNDERSAMPLING

1. Reduces Training Time

•By reducing the size of the dataset, undersampling can **speed up the training process** significantly, especially for models that are computationally intensive (like deep learning models).

2. Prevents Overfitting

•Since undersampling reduces the amount of data, it **prevents the model from memorizing the majority class samples**. This is particularly useful when the dataset is very large and overfitting is a concern.

ADVANTAGES OF UNDERSAMPLING

3. Improves Minority Class Detection

By balancing the class distribution, models are forced to pay **more attention to the minority class**, which can improve metrics like **Recall and F1-Score**.

4. Useful in Real-Time Systems

When quick model updates are needed, such as in streaming data or real-time applications, undersampling can be a practical approach to **keep the dataset size manageable**.

DISADVANTAGES OF UNDERSAMPLING

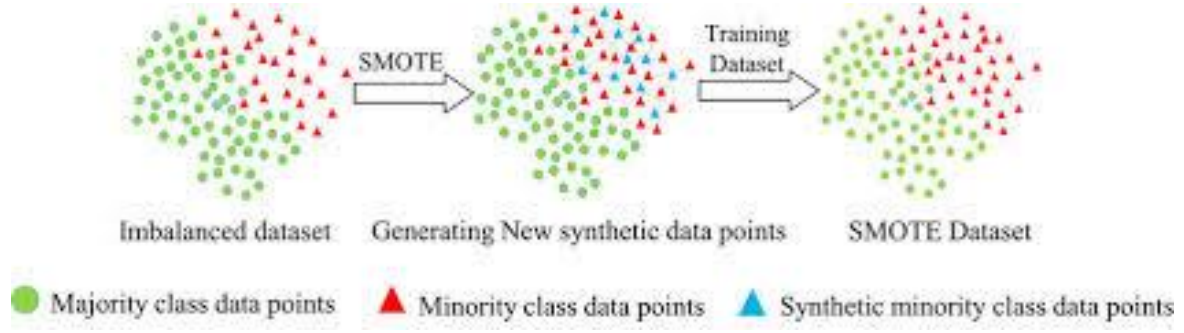
1. Loss of Information
The most significant drawback is that undersampling involves **removing potentially valuable data** from the majority class, which can lead to a loss of important patterns and features.

2. Not Suitable for Small Datasets
In scenarios where the dataset is already small, undersampling can worsen the problem by reducing the data to a point where the model **lacks sufficient information** to learn effectively.

3. Reduced Generalization Ability
Models trained on undersampled data may not generalize well to **real-world data**, which often retains the original imbalanced distribution. This can lead to poor performance when deployed on new, unseen data.

OVERSAMPLING

Oversampling balances the dataset by **increasing the number of examples in the minority class**.



Common Techniques for Oversampling

- ✓ **Random Oversampling:**

  Randomly duplicates samples from the minority class.

- ✓ **SMOTE (Synthetic Minority Over-sampling Technique):** Creates synthetic samples by interpolating between existing minority class samples.

- Calculates the difference between a sample and its nearest neighbor.
- Multiplies the difference by a random number between 0 and 1.
- Adds the difference to the sample

ADVANTAGES OF OVERSAMPLING

1. Balances Class Distribution

•Oversampling helps balance the dataset by increasing the number of examples in the minority class. This prevents the model from being biased towards the majority class.

2. Preserves Important Information

•Unlike **undersampling**, which reduces the dataset size by removing majority class samples, oversampling **retains all the original data**, ensuring that no potentially useful information is lost.

ADVANTAGES OF OVERSAMPLING

3. Class Performance

•Models trained on oversampled data are often better at identifying minority class samples, leading to improvements in metrics like **Recall, F1-Score, and Precision**, which are crucial for problems where detecting minority cases is important (e.g., fraud detection, medical diagnosis).

4. Useful When Data is Limited

•In cases where data collection is costly or difficult (e.g., rare diseases, cyber threats), oversampling is a practical way to **boost the sample size** of the minority class without requiring new data.

DISADVANTAGES OF OVERSAMPLING

1. Increases Risk of Overfitting

A major drawback of simple oversampling methods (like **random oversampling**) is that it **duplicates existing samples**, which can lead to overfitting, especially in small datasets. The model might learn to memorize these duplicates rather than generalizing patterns.

2. Longer Training Times

Since oversampling increases the dataset size, it can **significantly slow down the training process**, particularly for large and complex models (e.g., deep learning networks).

DISADVANTAGES OF OVERSAMPLING

3. Synthetic Data May Not Accurately Represent Reality

Techniques like **SMOTE** generate synthetic samples by interpolating between existing ones. These synthetic samples may not fully capture the underlying distribution of the data, which can introduce noise and reduce model accuracy.

4. Class Imbalance in Real-World Data

Oversampling can artificially balance the data, but this might not reflect the actual distribution in real-world scenarios. Models trained on oversampled data might not perform well when deployed on real, highly imbalanced data.

COMPARISON OF UNDERSAMPLING VS. OVERSAMPLING

Feature	Undersampling	Oversampling
Dataset Size	Reduced	Increased
Risk of Overfitting	Low	High (if duplicating samples)
Information Loss	Yes, removes data	No, retains original data
Use Case	Large datasets with a high imbalance	Small datasets, or when preserving all data is crucial
Algorithms	Random Undersampling, Cluster Centroids	SMOTE, ADASYN, Random Oversampling

Thank You ☺