MACHINE LEARNING AND NEURAL NETWORK

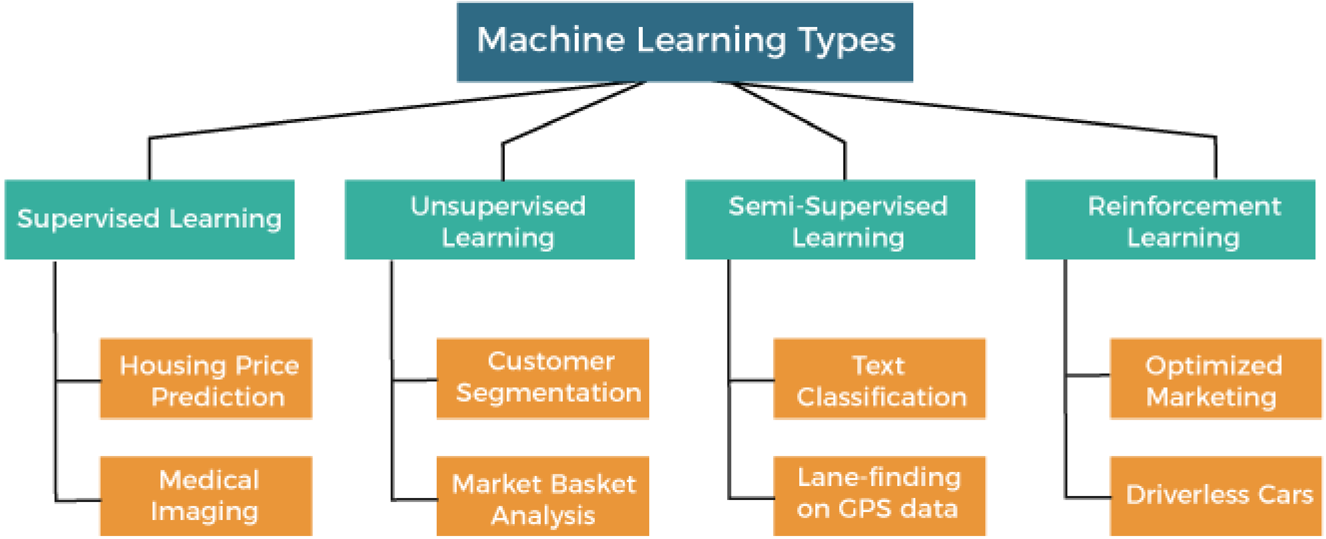
Lecture: Semi-supervised learning - Self Training

Annu Thomas (23019252)

- “Machine learning is a subfield of artificial intelligence (AI) that focuses on developing algorithms and statistical models that enable computers to "learn" from data, without being explicitly programmed.”

- “The primary objective of machine learning is to build models capable of making accurate predictions or decisions based on input data.”

- Machine learning techniques can be categorized into four types based on the structure of the algorithm used: Supervised learning, Unsupervised learning, Semi-supervised learning and Reinforcement learning.

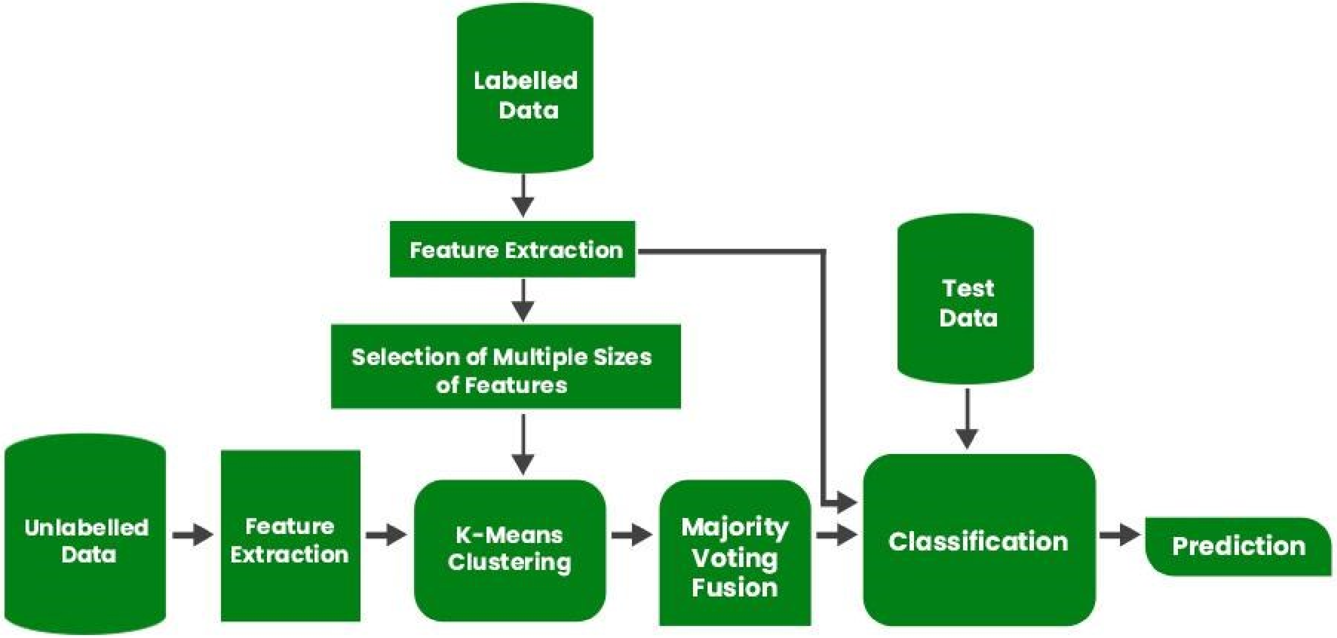


Semi-Supervised learning

“Semi-supervised learning is a machine learning technique that leverages a small amount of labelled data combined with a large amount of unlabelled data to train models.”

- It approach that bridges the gap between supervised and unsupervised learning. approach that bridges the gap between supervised and unsupervised learning.
- It combines elements from both supervised learning (using labelled data) and unsupervised learning (utilizing unlabelled data).
- Positioned as a middle ground, it benefits from the advantages of both learning paradigms to enhance model performance.
- Makes efficient use of the vast amounts of unlabelled data, which is often easier and cheaper to collect compared to labelled data.
- Reduces the need for extensive manual labelling, making it a cost-effective solution for industries where labelling is labor-intensive or expensive.
- The technique allows models to achieve better performance with less labelled data, thereby improving learning efficiency and reducing training time.
- Especially useful in real-world scenarios where labelled data is limited, but large amounts of unlabelled data are readily available.
- Models often improve iteratively by labelling new data points during training, leading to continuous performance enhancements.

Semi-Supervised Learning Workflow Diagram



Step 1: Input Data Sources

- Labelled Data:
  A small set of data with corresponding labels is used to guide the learning process.
- Unlabelled Data:
  A large set of data without labels provides additional information about the structure of the data distribution.

Step 2: Feature Extraction

- For both labelled and unlabelled data, feature extraction is performed.
- This step involves identifying relevant features or attributes from the raw data to serve as inputs for further processing.

Step3: K-Means Clustering (For Unlabelled Data)

- The extracted features from the unlabelled data undergo K-means clustering, a common unsupervised method used to group similar data points.
- This step helps in discovering hidden patterns and structures within the unlabelled dataset.

Step 4: Selection of Multiple Sizes of Features

- Various feature sizes are selected to ensure that the model can capture patterns at different levels of granularity.
- This helps improve the robustness and adaptability of the model.

Step 5: Majority Voting Fusion

- The output of the clustering process and the feature selection is combined using a majority voting mechanism.
- This fusion step consolidates multiple predictions or groupings into a unified representation, improving decision-making.

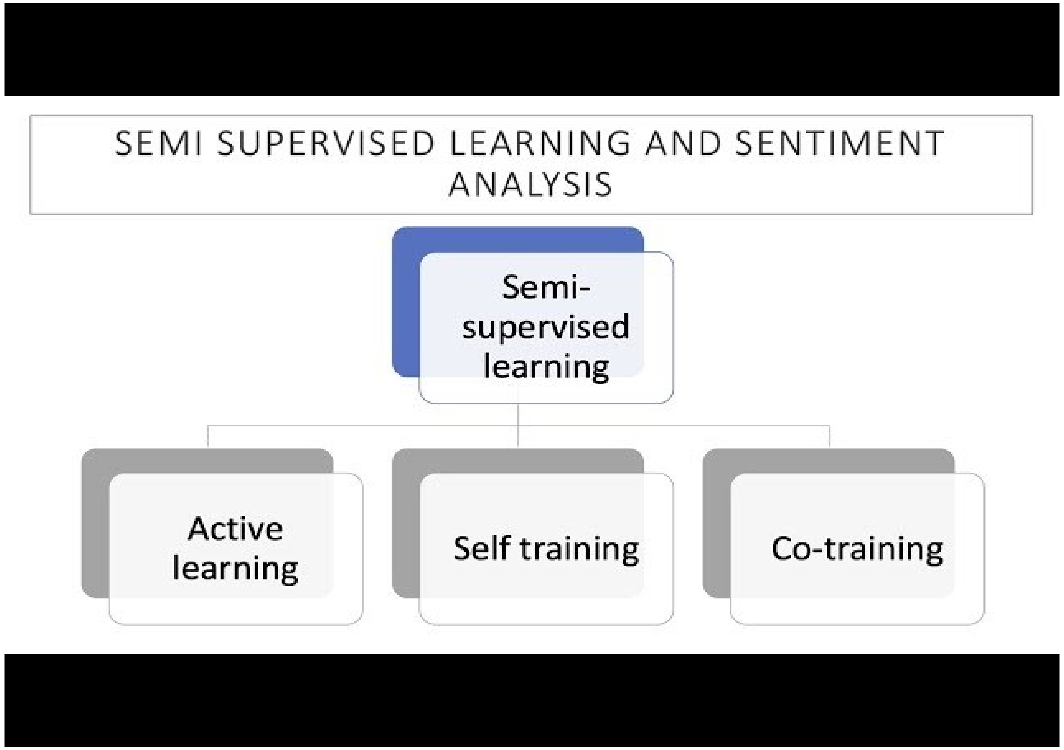
Step 6: Classification

- A classification model is trained using both the labelled data and the fused features from the unlabelled data.
- The test data is also fed into this classification model to make predictions.

Step 7: Prediction

- Finally, the classification model provides predictions for the test data.
- The semi-supervised approach ensures improved accuracy by leveraging the synergy between labelled and unlabelled data.

Methods of Semi-Supervised Learning



In this lecture we are discussing about what self-training is, its real-world application also its limitations.

Self-training

The self-training method is a straightforward yet effective technique in semi-supervised learning, typically involving the following steps:

Step 1: Initial Model Training

- A classifier is first trained on a small set of labelled data, similar to supervised learning. The model learns the patterns from this limited labelled set.

Step 2: Prediction on Unlabelled Data

- After training, the model is used to predict labels for the large pool of unlabelled data. The goal is for the model to apply the knowledge gained from the labelled set to classify the unlabelled examples.

Step 3: Selection of Confident Predictions

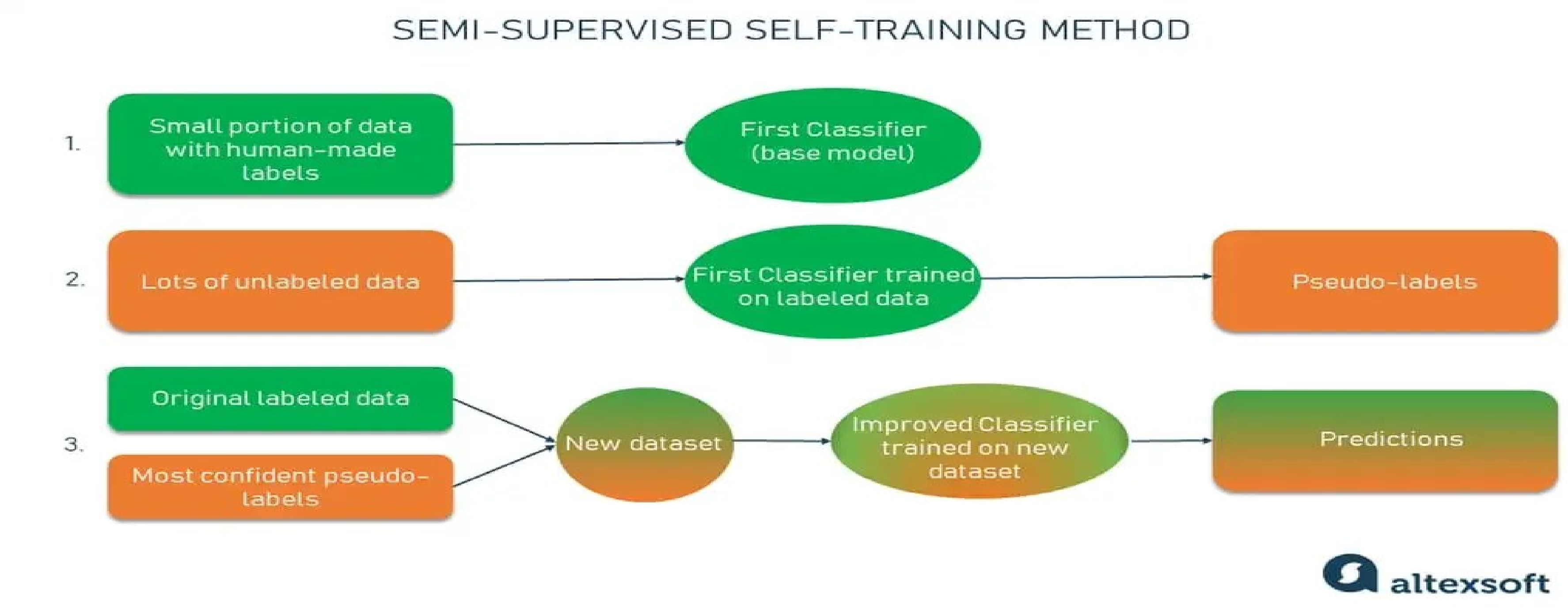
- Only those predictions for which the model has high confidence (i.e., those with the highest predicted probability) are selected for labelling.

Step 4: Pseudo-Labelling

- These confident predictions, known as pseudo-labels, are then treated as true labels and added to the training set. The model is then retrained using both the original labelled data and the newly pseudo-labelled data.

Step 5: Iterative Process

- This cycle repeats itself, with each iteration refining the model's predictions on the unlabelled data, gradually increasing the amount of labelled data the model learns from. Over time, this iterative process improves the model's performance by utilizing both labelled and unlabelled data.


SEMI-SUPERVISED SELF-TRAINING METHOD
1.
Small portion of data with human-made labels
First Classifier (base model)
2.
Lots of unlabeled data
First Classifier trained on labeled data
Pseudo-labels
3.
Original labeled data
Most confident pseudo-labels
New dataset
Improved Classifier trained on new dataset
Predictions
altexsoft

Benefits Of Self-Training:

Effective Utilization of Unlabeled Data

- Self-training harnesses the potential of vast unlabeled datasets by assigning pseudo-labels, making use of data that would otherwise remain untapped.
- It gradually refines the model's performance by incorporating confidently labeled data into the learning process.

- This approach maximizes the value of available data, even with a limited amount of labeled examples.

Simplicity and scalability

- Self-training is one of the simplest semi-supervised learning methods, requiring no complex algorithms or additional resources.
- The process remains consistent regardless of dataset size, making it scalable for both small datasets and massive collections of data.
- It can work seamlessly with various machine learning models, making it versatile for different data sizes and domains.

Flexibility Across Models

- Self-training is model-agnostic, meaning it can be applied to any learning algorithm, from logistic regression and decision trees to deep neural networks.
- This adaptability makes it suitable for a wide range of tasks, including image classification, sentiment analysis, and text processing.
- Its compatibility with diverse models allows for easy integration into existing machine learning pipelines.

Examples of Self-Training in Semi-Supervised Learning:

- **Text Classification (Spam Detection):**

  Scenario:

Imagine you are working for a messaging platform that handles millions of SMS messages daily. Your task is to build a spam detection system to filter out unwanted or harmful messages automatically. However, manually labelling every single message as spam or ham is impractical due to the sheer volume, cost, and time it would require.

Key Challenges:

Massive Unlabelled Dataset: Most of the available SMS data lacks labels indicating whether a message is spam or not.

High Cost of Manual Labelling: Labelling large datasets manually is expensive and time-consuming.

Need for Accurate Models with Minimal Labels: An efficient model is required to achieve high accuracy without relying on fully labelled data.

The Solution:

To overcome these challenges, you apply semi-supervised learning with a Self-Training Classifier using the SMS Spam Collection Dataset. Here's how the solution is implemented:

1. Limited Labelled Data: Begin with only a small subset of the dataset (around 10%) labelled as ham or spam.
2. Utilize Unlabelled Data: The remaining unlabelled data is leveraged to enhance the model’s learning process.
3. Self-Training Process: The model is initially trained on the labelled data, predicts labels for the unlabelled data, and iteratively incorporates its most confident predictions to expand the training set.
4. Evaluation and Results: The model is tested on a separate set of messages to assess its accuracy and effectiveness.

- **Sentimental Analysis:**

  Scenario:

Imagine you're tasked with developing a sentiment analysis model for a social media analytics company. The objective is to classify tweets or comments as either positive or negative based on their content. However, you're facing a common challenge:
Limited Labelled Data: You only have a small set of tweets that have been manually labelled as either positive or negative.

Large Unlabelled Data: You have access to a vast collection of unlabelled tweets that would be beneficial for training the model, but labelling them manually would be time-consuming and costly.

Solutions:

You have only a small set of tweets that are manually labelled as positive or negative, which can hinder the performance of machine learning models since they typically require a substantial amount of labelled data to learn patterns effectively. To solve this:

1. Train Initial Model: Begin by training a classifier (e.g., Multinomial Naive Bayes) using the small labelled dataset (for instance, 10% of the total data).

- The small labelled dataset will help the initial model learn basic patterns in the text that associate certain words with positive or negative sentiments.

2. Pseudo-Label Unlabelled Data: Use the trained model to predict labels for the unlabelled data (which makes up the majority of the dataset). These predictions (pseudo-labels) are then treated as if they were labelled data.

- By doing this, you increase the size of your training data without the need for manual labelling.

3. Iterative Refinement (Self-Training):

- The Self Training Classifier allows for iterative learning, where the model first trains on the labelled data, then uses its own predictions to label the unlabelled data, and continues to refine itself by re-training with the newly pseudo-labelled data.

We have a vast collection of unlabelled tweets, which are valuable for training the model, but manually labelling them would be costly and time-consuming. To solve that:

1. Vectorize the Text: Convert the text data into numerical features using a CountVectorizer or TF-IDF Vectorizer. This step transforms each tweet into a feature vector representing word frequencies or weighted term frequencies.

- Apply the same vectorizer to both labeled and unlabeled datasets for consistency.

2. Merge Labeled and Unlabeled Data:

- Combine both the labeled and unlabeled data into a single training set.
- In the semi-supervised approach, treat the unlabelled data as pseudo-labelled (using predicted labels).

3. Train the Self-Training Classifier:

- Use the SelfTrainingClassifier (with a base classifier like Multinomial Naive Bayes) to train the model on both the labelled and pseudo-labelled data.
- The model learns from both the original labelled data and the newly pseudo-labelled data, allowing it to improve its accuracy.

4. Maximize Data Use:

- By leveraging both labelled and unlabelled data, you make the most of all available data, reducing the need for manual labelling while improving the model's generalization ability.

In this lecture example, we demonstrate how to use semi-supervised learning to classify sentiment with minimal labelled data. The Self-Training Classifier allows the model to leverage unlabelled data, improving performance while reducing the need for extensive manual labelling. This technique is particularly useful in real-world scenarios where data labelling is expensive or impractical.

Drawbacks of Self Training Learning

- Error Accumulation: Early incorrect pseudo-labels can propagate, reducing overall model accuracy.
- Dependence on Initial Classifier: Success relies heavily on the initial model’s performance; a weak starting model can lead to poor outcomes.
- Limited Applicability: Self-training may struggle if the distribution of unlabeled data differs significantly from the labeled set.
- Imbalance Issues: It tends to favor the majority class, especially when the labeled dataset is small and unbalanced.
- Risk of Overfitting: The model may overfit to the pseudo-labeled data if it lacks sufficient diversity.
- Iteration Challenges: Deciding the optimal number of iterations can be difficult and impacts performance significantly.
- Threshold Sensitivity: Setting appropriate confidence thresholds for pseudo-labeling requires careful tuning and expertise.

Reference

1. Bai, T., Zhang, Y. and Zong, C., 2020. Semi-supervised learning by self-training. *Information Sciences*, 517, pp.44-58. Available at: https://doi.org/10.1016/j.ins.2020.02.049.

2. Lee, D.H., 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proceedings of the 30th International Conference on Machine Learning*. Available at: https://arxiv.org/abs/1308.1145.

3. Zhao, C., Zhang, Z. and Chen, J., 2023. Graph-based self-training for semi-supervised deep similarity learning. *Sensors*, 23(8), p.3944. Available at: https://doi.org/10.3390/s23083944.

4. Triguero, I., Gonzalez, S. and Herrera, F., 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42(2), pp.245-284. Available at: https://doi.org/10.1007/s10115-013-0706-y.

5. van Engelen, J.E. and Hoos, H.H., 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2), pp.373-440. Available at: https://doi.org/10.1007/s10994-019-05855-6.