Classification and Prediction

Flavius Runceanu - Arinze Emeti

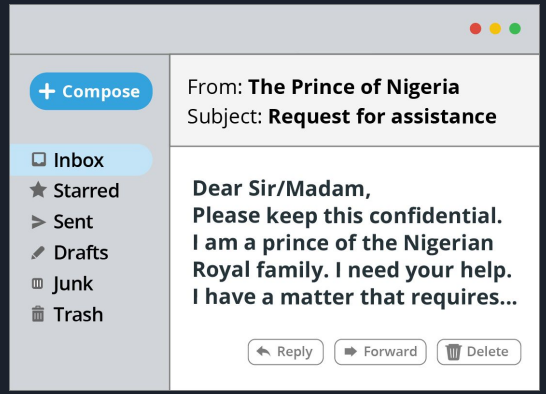
SVM - Support Vector Machines - Real Use Cases

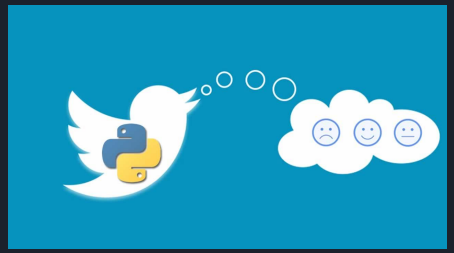
Email Spam Detection: classify emails as spam or non-spam, improving filtering systems.

Image Classification: recognize objects and patterns in images, enabling accurate classification.

Text Categorization: categorize documents based on content, aiding in tasks like sentiment analysis.

Financial Forecasting: predict stock prices and currency exchange rates.





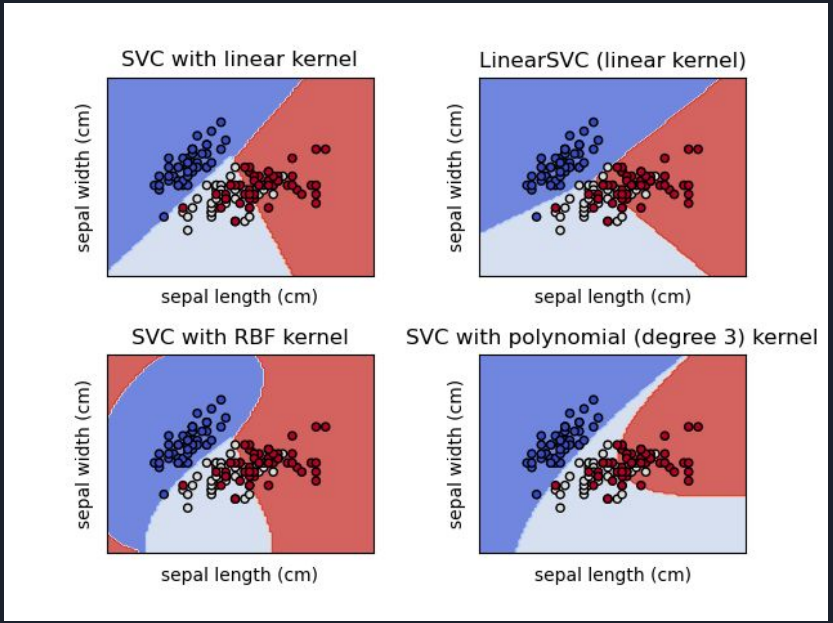
SVM - Support Vector Machines

- Support Vector Machines (SVMs) are powerful supervised learning algorithms used for *classification* and *regression* tasks.
- They aim to find an *optimal hyperplane* that *separates data points of different classes* with the largest possible margin.

In some cases, the classes in the input data cannot be separated by a straight line (hyperplane) in the original feature space. This is known as linear inseparability.

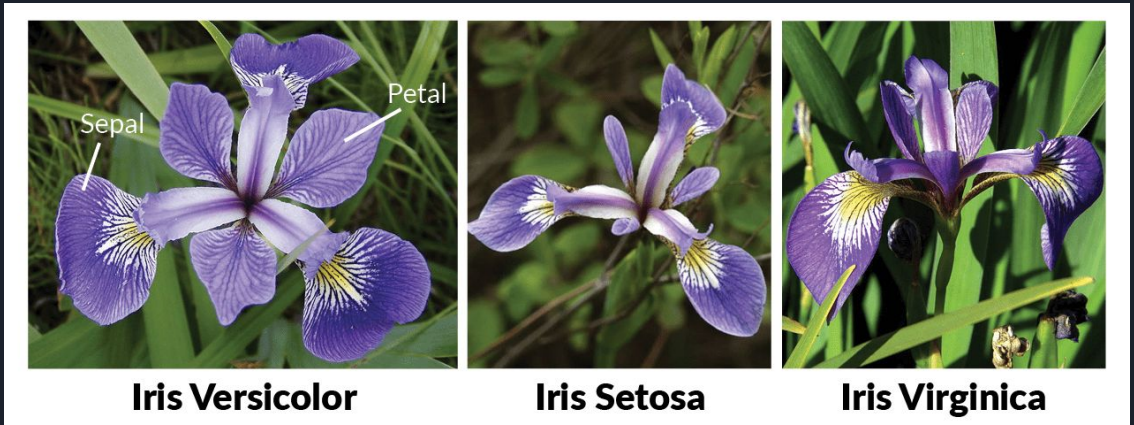
Kernels address the problem of linear inseparability by applying a mathematical function that maps the original input data into a higher-dimensional feature space.

The kernel function computes the dot product between two data points in the new feature space, *without explicitly transforming the data.*



SVM - Support Vector Machines

Iris Database Demo



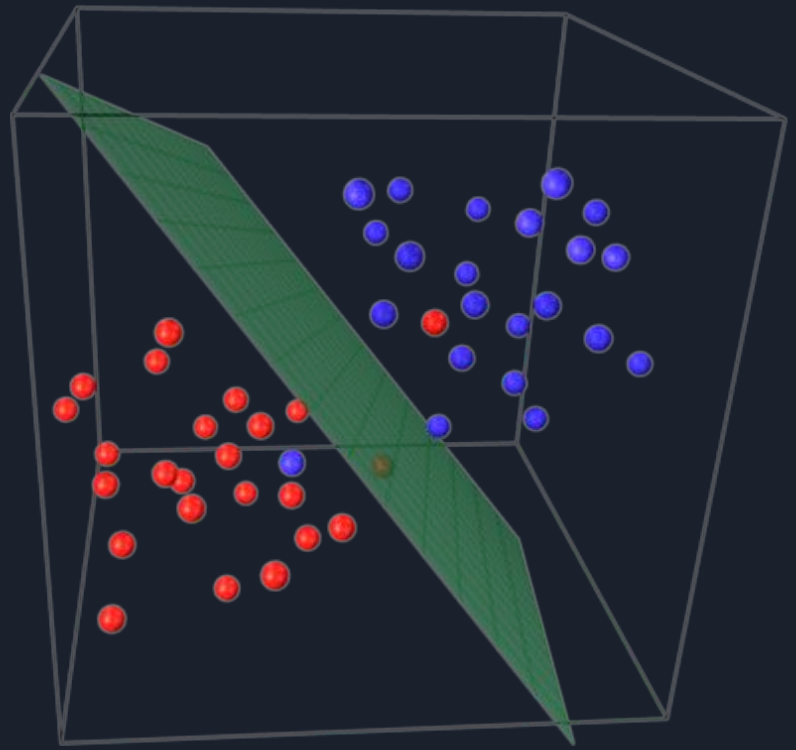
SVM - Advantages and Disadvantages

Advantages:

1. Effective in high-dimensional spaces
2. Robust against overfitting
3. Versatile for different data types
4. Handles non-linear decision boundaries

Disadvantages:

1. Computationally intensive
2. Sensitivity to parameter tuning
3. Difficulty with large datasets
4. Lack of interpretability



GCN - Graph Convolutional Networks - Real Use Cases

Social Network Analysis: Uncover community structures, identify influential nodes, and predict link formations in social networks.

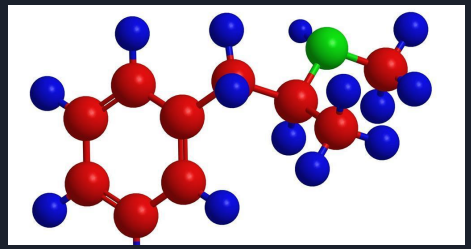
Drug Discovery: Analyze molecular structures, predict chemical properties, and accelerate drug development.

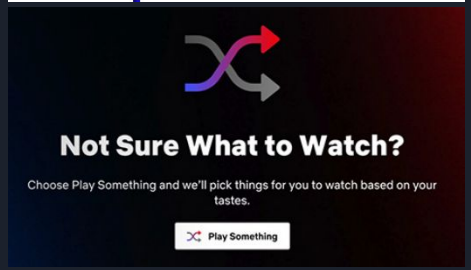
Fraud Detection: Detect fraudulent activities in financial transactions and online platforms.

Knowledge Graph Completion: Infer missing relationships in knowledge graphs for knowledge representation and semantic search.

Traffic Flow Prediction: Predict traffic flow and optimize transportation systems.

Recommendation Systems: Provide personalized recommendations based on user preferences and item characteristics.





GCN - Graph Convolutional Networks

- GCNs are deep learning models designed for graph-structured data.
- They enable tasks like node classification, link prediction, and graph-level regression.

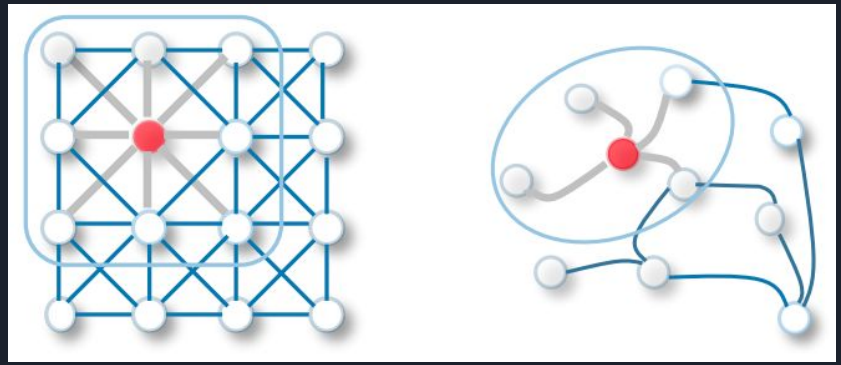
Graph Representation: GCNs operate on graphs with nodes and edges, where nodes may have features.

Graph Convolution: GCNs perform convolutions on graphs, capturing local and global relationships.

Message Passing: GCNs exchange information between nodes through graph edges.

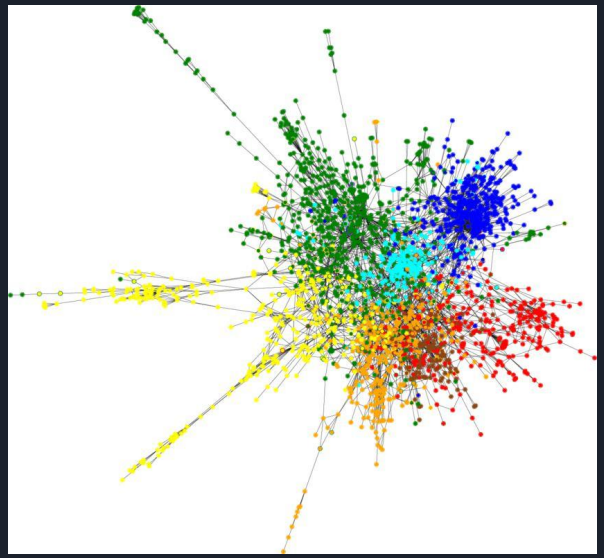
Graph Aggregation: GCNs aggregate information from nodes to obtain a global representation of the graph.

CNNs are specially built to operate on regular (Euclidean) structured data, while GNNs are the generalized version of CNNs where the numbers of nodes connections vary and the nodes are unordered.



GCN - Graph Convolutional Networks

Cora Database Demo



1. **Case_Based:** Publications related to case-based reasoning.
2. **Genetic_Algorithms:** Publications related to genetic algorithms and evolutionary computation.
3. **Neural_Networks:** Publications related to neural networks and deep learning.
4. **Probabilistic_Methods:** Publications related to probabilistic methods and Bayesian approaches.
5. **Reinforcement_Learning:** Publications related to reinforcement learning and decision-making processes.
6. **Rule_Learning: Publications** related to rule learning and inductive logic programming.
7. **Theory:** Publications related to theoretical aspects of computer science.

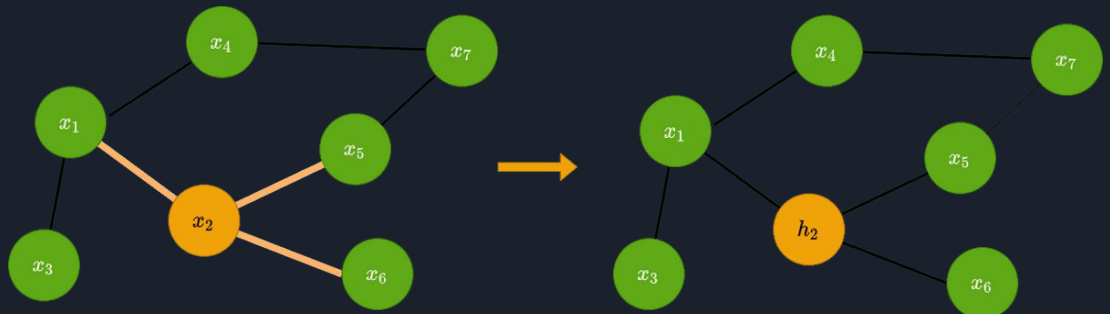
GCN - Advantages and Disadvantages

Advantages:

1. Captures **complex relationships** in graph data.
2. Handles varying node connections and unordered nodes.

Disadvantages:

3. **Computationally intensive**, especially for large graphs.
4. **Limited interpretability** due to complex operations on graph structures.
5. Challenges with **dynamic or changing graph structures.**



$$h_2 = g(x_1, x_5, x_6)$$

References:

Support Vector Machines (SVMs):

1. "A Gentle Introduction to Support Vector Machines" by R. Berwick: https://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf
2. "Support Vector Machines" by Andrew Ng: https://www.youtube.com/watch?v=_PwhiWxHK8o
3. "Support Vector Machines" by Chris McCormick: https://mccormickml.com/2014/03/04/support-vector-machines-tutorial/

Graph Convolutional Networks (GCNs):

1. "Graph Convolutional Networks" by Thomas Kipf: https://tkipf.github.io/graph-convolutional-networks/
2. "Graph Convolutional Networks: A Comprehensive Review" by Zonghan Wu et al.: https://arxiv.org/abs/1901.00596
3. "Graph Convolutional Networks for Node Classification" by Michael Bronstein: https://www.youtube.com/watch?v=1-XysCimDiM