ML Roadmap Knowledge Graph

Step 1 High level notes for building the Mind Map

Foundations:

- **Math:** Linear algebra, calculus, probability, and statistics.
- **Programming Languages:** Python, R, SQL.

Learning Approaches:

- **Supervised Learning:** Regression, classification.
- **Unsupervised Learning:** Clustering, dimensionality reduction.
- **Reinforcement Learning:** Agents, environments, reward functions.

Techniques & Algorithms:

- **Neural Networks:** Feedforward networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), transformers.
- **Decision Trees:** Random forests, gradient boosting machines (GBMs).
- **Support Vector Machines (SVMs):** Kernels, margin optimization.

Tools & Frameworks:

- **TensorFlow:** Keras, TensorFlow Lite.
- **PyTorch:** TorchScript, PyTorch Lightning.
- **Keras:** Sequential API, functional API.

Practical Applications:

- **Natural Language Processing (NLP):** Tokenization, named entity recognition (NER), sentiment analysis.
- **Computer Vision:** Image classification, object detection, image segmentation.
- **Recommendation Systems:** Collaborative filtering, content-based filtering, hybrid methods.

Model Evaluation:

- **Metrics:** Accuracy, precision, recall, F1 score, ROC-AUC.
- **Validation Techniques:** Cross-validation, bootstrapping.

Data Engineering:

- **Data Cleaning:** Handling missing values, outliers, feature scaling.
- **Data Visualization:** Matplotlib, Seaborn, Tableau.

Deployment:

- **Model Serving:** REST APIs, cloud deployment (AWS, GCP, Azure).

- **Monitoring:** Performance monitoring, drift detection

Basic Mind Map outline to begin building

Foundations

/ \

Learning Approaches Tools & Frameworks

/ | \ / | \

Supervised Unsupervised Reinforcement TensorFlow PyTorch Keras

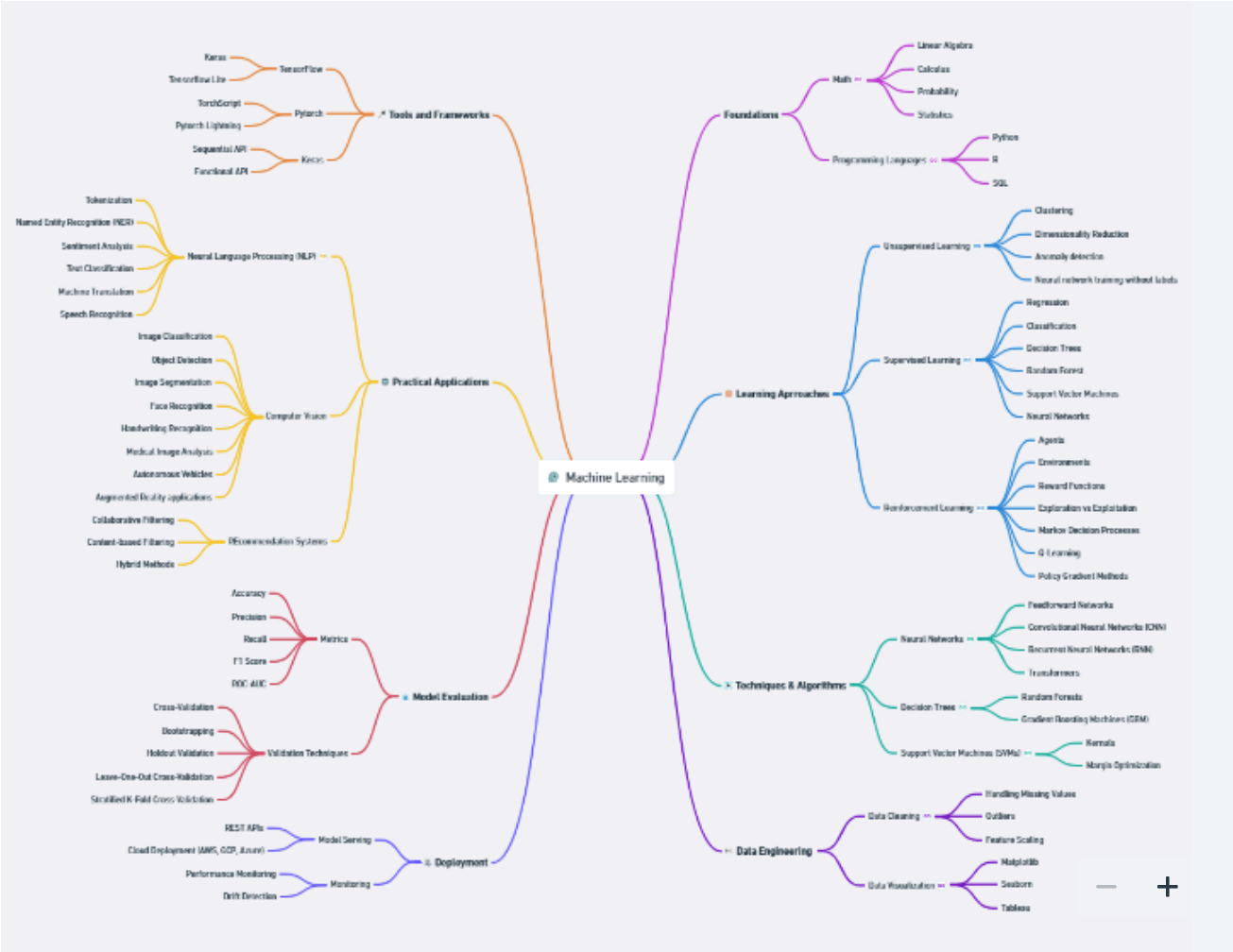
| | | | | |

Algorithms Algorithms Algorithms Algorithms Algorithms Algorithms

| / / | | |

Practical Applications -- Model Evaluation -- Data Engineering -- Deployment

Created Mind Map: ML Roadmap Knowledge Mind Map



Report:

Through this exercise we constructed a knowledge graph using Daniel Bourke's Machine Learning Roadmap as our foundation. The activity was designed to expand our understanding of AI/ML by identifying primary areas and mapping their linked concepts.

The following steps were implemented to construct the knowledge graph.

1. Account Setup: Created a new Whimsical account.
2. Conducted an extensive review of Daniel Bourke's Machine Learning Roadmap.
3. Note-Taking: We documented the main areas as Foundations plus Learning Approaches along with Techniques & Algorithms and Tools & Frameworks then Practical Applications.
4. Graph Creation: We used Whimsical to build a mind map that displayed major areas as primary nodes while linking secondary nodes to represent subtopics and particular concepts.
5. Collaboration: Discussions were conducted to establish topic connections and the graph was organized based on them.

Challenges Several challenges arose during this process:

1. Technical Issues: At the start I encountered challenges understanding how to navigate Whimsical so I had to conduct further research.
2. Conceptual Understanding: The process of mapping connections between concepts proved difficult particularly when establishing dependencies and relationships between them.

The exercise significantly enhanced our grasp of artificial intelligence and machine learning concepts. Our understanding revealed that the network of machine learning concepts spanning foundational knowledge through to practical applications proved more complicated than we originally expected. Here are some specific insights:

1. Interconnectedness of Concepts: The exercise demonstrated that basic knowledge in mathematics and programming serves as the essential foundation for advanced subjects such as neural networks and computer vision. Our understanding of these connections made us recognize the crucial role that a strong foundation plays.
2. Practical Applications: The application of various learning methods and algorithms to tangible real-world situations such as NLP and computer vision demonstrated how theoretical knowledge translates into practical use.
3. Collaboration: The collaborative effort on the knowledge graph led to multiple perspectives and enhanced collective understanding. Our critical evaluation skills and capacity to synthesize information improved through the discussion of various viewpoints and establishment of connections.
4. Tool Proficiency: We improved our ability to use mind-mapping tools after we got through the early difficulties with Whimsical and this new skill will help us with upcoming projects.

5. Conceptual Clarity: Our understanding and memory of information improved when we visualized complex subjects as structured maps.

The exercise provided substantial educational value and numerous benefits. The exercise demonstrated how essential it is to adopt a structured method for learning and grasping AI/ML concepts. For future iterations, we recommend the following:

1. Tutorials for Tools: A short tutorial for using tools like Whimsical will make processes more efficient while minimizing beginner frustration.
2. Incremental Learning: Segmenting complex ideas into smaller sections promotes enhanced comprehension and effective mapping.
3. Continuous Improvement: The enhancement of understanding and accuracy can be achieved through ongoing feedback and iterative improvements to the knowledge graph.
4. Integration with Real-World Examples: The use of real-world case studies and examples helps students understand theoretical concepts by providing practical applications.

The exercise enhanced our understanding of AI/ML while simultaneously strengthening our collaboration abilities and critical thinking skills and our proficiency with visualization tools. Our learning exercise established a robust base for continued education and exploration in artificial intelligence