Advancing Decision Science: Lessons from Machine Learning

Bridge on Constraint Programming and Machine Learning

Serdar Kadıoğlu

AI Center of Excellence, Fidelity Investments

Adj. Assoc. Prof, Dept. of Computer Science, Brown

skadio.github.io

Learning & Reasoning

Data Science: ML/DL/LLMs/etc

Focuses on **machine learning using historical data** to identify patterns and make predictions. Excels at pattern recognition, classification, and forecasting.

Predictive Models

- Learning from historical data patterns

- Probabilistic predictions and insights

- Ideal for unstructured problems

- Applications include recommendation systems, image recognition, and natural language processing

Decision Science: SAT/CP/MP/LS/etc

Focuses on **combinatorial satisfaction and optimization** using logical and mathematical models. Provides provable optimality and explicit reasoning.

Prescriptive Models

- Mathematical and logical formulations

- Provably optimal solutions

- Perfect for structured problems

- Applications include verification, planning, scheduling, routing, and resource allocation

The Software Gap in Decision Science

Success of Data Science

The Learning community has benefited tremendously from **open-source software** such as PyTorch, and TensorFlow, along with established **packages** like NumPy, SciPy, and scikit-learn.

Opportunity for Decision Science

The Reasoning community trails behind in establishing a **similar software ecosystem**, hindering widespread adoption and collaboration of reasoning solvers & software, limiting its potential and impact.

Bridging the Gap

Let's explore **concrete examples** from the ML & Data Science community and consider their equivalents or similar versions for the Decision Science community.

1 **Ease of Use and Accessibility**

2 **Integrated Environments**

3 **Community Driven Development**

4 **Showcase Success**

Bridging the Gap: Ease of Use

One-Click Installation

Simple **pip install** command for all decision science tools

No Dependency Hassles

Eliminate complex **dependency management** and version conflicts

Quick Start Experience

Making it easier for **new users** while supporting experienced developers

Ex: CPMPy, Pyomo, PySAT, SCIP, OR-Tools, ALNS, PyVRP, PyJobShop, DIDPpy, ...

Seq2Pat

Sequence to Pattern generation for constrained pattern mining

Discover frequent sequential patterns in large sequence databases with constraints

Collaboration with CMU
Kadıoğlu et. al. (AI Magazine'23)

```python
# Example to show how to find frequent sequential patterns
# from a given sequence database subject to constraints
from sequential.seq2pat import Seq2Pat, Attribute

# Seq2Pat over 3 sequences
seq2pat = Seq2Pat(sequences=[["A", "A", "B", "A", "D"],
                             ["C", "B", "A"],
                             ["C", "A", "C", "D"]])

# Price attribute corresponding to each item
price = Attribute(values=[[5, 5, 3, 8, 2],
                          [1, 3, 3],
                          [4, 5, 2, 1]])

# Average price constraint
seq2pat.add_constraint(3 <= price.average() <= 4)

# Patterns that occur at least twice (A-D)
patterns = seq2pat.get_patterns(min_frequency=2)
```

pip install seq2pat

35K+ downloads

Bridging the Gap: Ease of Use

Ready-to-Run Environments

Docker containers come fully configured with all necessary dependencies installed and ready to use

Instant Setup

Start solving problems **immediately** without dealing with complex installation processes

Cross-Platform Consistency

Ensure **consistent behavior** across different platforms and operating systems

Built-in Resources

Access **pre-loaded** example problems and documentation right out of the box

Ex: Dockerized SCIP for Teaching

Bridging the Gap: Ease of Use

1 **Standardized Interfaces**

Common APIs for solvers and models that provide a unified experience across different tools.

Reduce the learning curve when switching between solvers and enables easier integration with existing codebases and flows.

2 **Interactive Documentation**

Create comprehensive, interactive tutorials and documentation with real-time examples and **notebooks**.

Users understand concepts quickly and provides practical implementation guidance for common use cases.

3 **Automated Model Validation**

Built-in tools for **model validation** and **debugging** help users identify common modeling errors .

Reduce friction of working with complex constraint models and speeds up development time.

fit + predict ≈ model + solve

Bridging the Gap: Accessibility

(1) Low Code/No Code Environments

Intuitive drag-and-drop interfaces for model building that enable optimization models without writing code.

Visual model validation, pre-built components for common constraints, and the ability to import data directly from business databases.

Ex: AIMMS Modeling Environment, IBM Decision Studio, etc.

(2) Modeling Assistants

Modeling Co-Pilots to suggest model improvements, identify potential issues, and help translate business requirements into mathematical constraints.

Learn from existing models to recommend best practices, detect common modeling patterns, and generate initial structures based on natural language.

Ex: Holy Grail 2.0, ChatOpt, OptiGuide, OptiMUS, MeetMate, MCPSolver, etc.

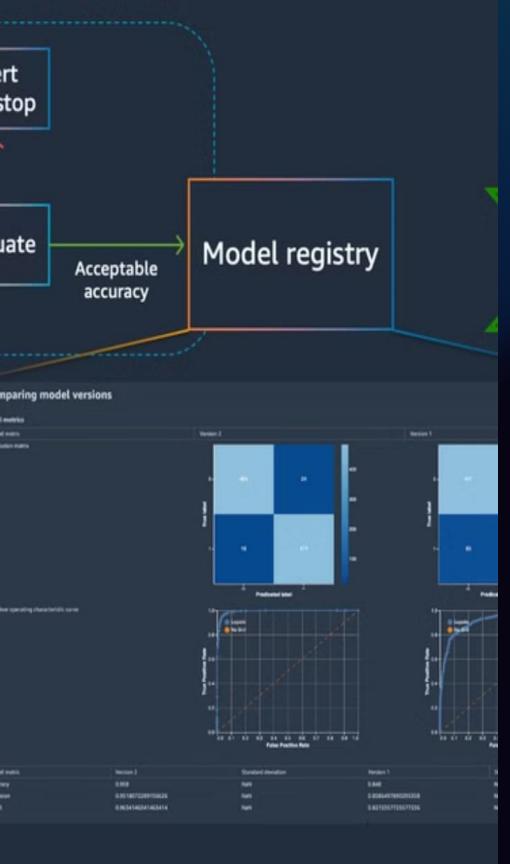# Integrated Data & Model Cards

① Purpose

Increase **transparency** and accountability in dataset use by providing detailed documentation.

② Content

Includes dataset **overview**, collection methods, intended use, ethical considerations, and limitations.

Includes **model details**, intended use, performance **metrics**, ethical considerations, and limitations.

③ Benefits

Promotes transparency, accountability, and **Responsible AI** development.

Model Registry & DecisionOps

1 **Version Control**

Keeps track of different **versions of models,** allowing management of updates and changes over time.

2 **Model Observability**

Stores important **information** about each model, such as parameters, training data, and performance metrics.

3 **Collaboration**

Facilitates collaboration among data scientists and engineers by providing a shared space to access and manage models.

Ex: NextMv, etc.

Decision Science as a Service

1 **Cloud-based Tools**

Provide **scalable infrastructure** for building and deploying optimization models.

2 **API Integration**

Enable seamless **API integration** with existing systems and workflows.

3 **Managed Services**

Maintenance and updates for decision science tools.

Ex: TimeFold Planning AI, Probabl, Google Routing API, MathOpt API

1 **Prototype**
Develop initial models

2 **Scale**
Adapt for large-scale data

3 **Deploy**
Integrate with existing systems

4 **Monitor**
Continuous eval and improve

Hugging Face & Community

Transformers Library

Popular library for natural language processing with **pre-trained models.**

Model Hub

Platform for sharing and discovering pre-trained models.

Datasets

Offers a wide range of **datasets** for machine learning tasks.

Ex: Ner4Opt fine-tuned transformer models on optimization corpora

Is there a Hugging Face for Decision Science?

1 **Model Hub**

Create a repository for sharing **pre-built constraint models**

2 **Collaborative Platform**

Foster **community-driven development** of decision science tools

3 **Easy Deployment**

Provide **infrastructure** for deploying decision models as web services

Ex: CoinOR founded in 2000 🤗

Demo Spaces

1 **Interactive ML Apps**

Supports building applications using **Streamlit, Gradio, Docker, or static HTML.**

2 **Hosting and Deployment**

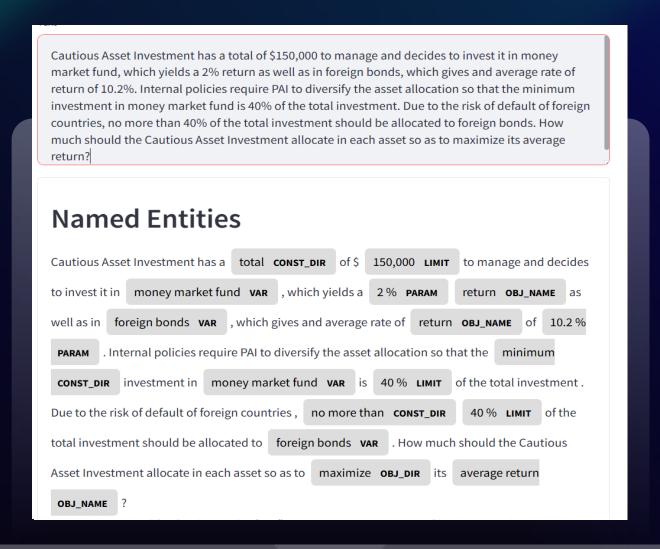Provides **easy deployment** options with various hardware choices.

3 **Collaboration and Sharing**

Enables **sharing ML projects** with the community and Git integration for version control.

Ner4Opt
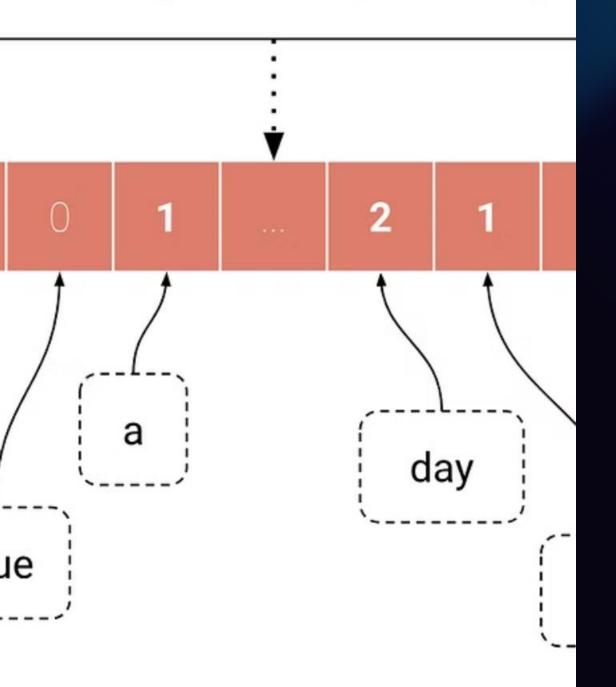
Named entity recognition for optimization

Extract variables, objective, parameters, and constraints from natural text

Kadıoğlu et. al. (Constraints'24)

pip install ner4opt

Ner4Opt Named Entity Recognition for Optimization Demo

Pre-Trained Embeddings

1 **Sentence Transformers**

Framework for generating **dense vector representations** for sentences, paragraphs, and images.

2 **Key Features**

Pre-trained models, custom training, similarity computation, and multilingual support.

3 **Applications**

Semantic search, paraphrase mining, clustering, and textual similarity.

OpenML

① Datasets

Hosts a variety of **uniformly formatted datasets** with rich, consistent metadata.

② Tasks

Defines **standardized tasks** for benchmarking and comparing different algorithms and models.

③ Experiments

Records exact datasets, algorithms, and library versions ensuring **reproducibility**.

Ex: MIPLIBing, DIMACS, CNF

```python
import openml

# List all datasets and their properties
openml.datasets.list_datasets(output_format="dataframe")

# Get dataset by ID
dataset = openml.datasets.get_dataset(61)

# Get dataset by name
dataset = openml.datasets.get_dataset('Fashion-MNIST')

# Get the data itself as a dataframe (or otherwise)
X, y, _, _ = dataset.get_data(dataset_format="dataframe")
```

Resources and Tools

1 **Community-Driven Development**

Collaborative development, maintenance, and events including hackathons and competitions. Mentorship and certification programs to validate skills

2 **Benchmarking & Leaderboards**

Publicly accessible datasets with data cards, models with model cards, **benchmarks, leaderboards**, evals.

3 **Educational Resources**

Publicly available **educational material** including interactive tutorials, blog posts, and online courses.

The CP-SAT Primer   LP Primer
Text2Zinc Cross-Domain Dataset & Leaderboard for LLM Modeling

Advanced Features and Integration

Explainability

Tools providing insights into optimization model decisions, similar to **Explainable AI** techniques.

Interactive Dashboards

Visualize and interact with optimization models and results in real-time.

Hybrid Software

Integrated tools for combining **learning and reasoning** models.

Automated Tuning

Hyperparameter optimization for best performance in optimization models.

1

2

3

4

Ecole Library Extensible Combinatorial Optimization Learning Environments
DSPy: Declarative Self-Programming for LLM Prompting

Showcasing Successful Applications

The Art of Possible

Demos and proof-of-concept application spaces to communicate the potential of decision science, similar to HuggingFace Spaces.

Industry Applications

Showcase successful industry applications with practical impact addressing unique needs of sectors such as healthcare, finance, and manufacturing.

Real-World Impact

Highlight how decision science solutions have improved efficiency, reduced costs, or solved complex problems in various industries.

Ex: INFORMS Prizes   Constraint Applications Blog   IAAI – Innovative Applications of AI

Opportunities to Advance Decision Science

Ease of Use

Easy installation, ready-to-run environments, and standardized interfaces.

Accessibility

Low code/no code environments and modeling assistants.

Constraint Model Registry

Hosting base constraint models for common applications.

Pre-trained Embeddings

Latent instance representations for CP, MIP, SAT.

Cloud Services and APIs

Online services, API integration, and emergence of DecisionOps.

Community-Driven Development

Collaborative development, events, mentorship, and certification programs.

Datasets and Leaderboards

Publicly accessible datasets and benchmarks for decision science problems.

Educational Resources

Interactive tutorials, blog posts, and online courses on decision science topics.

Successful Applications

Interactive tutorials, blog posts, and online courses on decision science topics.

Call to Action

1 **Contribute**

Participate in open-source development of decision science tools.

2 **Collaborate**

Build a robust ecosystem to share and collaborate.

3 **Improve Tech Stack & Innovate**

Enhance standards and user experience to attract new participants and foster collaboration.

skadio.github.io