ML pipeline exploration

Do it yourself: https://github.com/IBM/grammar2pddl

Tools and links for hands-on exercise

- HTN2PDDL translation: https://github.com/ronwalf/HTN-Translation
- K* planner: https://github.com/ctpelok77/kstar
- PlanUtils: https://github.com/AI-Planning/planutils
- PDDLEditor plugin for VSCode: https://marketplace.visualstudio.com/items?itemName=jan-dolejsi.pddl

Tools and Applications

AI Planning: Theory and Practice

Challenges of using AI planning in applications

- Modeling: Application domain ≠ planning domain
- There are very few integrated toolchains for AI planning
- There are few libraries/frameworks that integrate AI planners
- Round-tripping is cumbersome

Pre/post-processing

- What is the similarity/equivalence between what desired output and plans
  - Example: Are ML pipelines like plans and how? What are our ‘actions’?
- What are our inputs and is there a close-enough planning formalism?
  - Example: We have data science grammars, that look a lot like HTN specifications
- How do we map input elements to predicates and actions?
  - Example: grammar tokens = actions, grammar rules = methods
- Do we need ‘maintenance actions’ - how do we need to post-process plans?
  - Example: in the HTN to PDDL translation there are maintenance ‘method start’ and ‘method end’ actions that do not produce any pipeline tokens

AutoAI: Automating ML Pipeline Generation

Problem

Space of possible pipelines is huge

Humans can explore only a tiny portion of it

Humans are biased towards the pipelines they already are familiar with

Benefits

- Generation of pipelines of high accuracy automatically
- Reduces the need for human data scientists

Solution

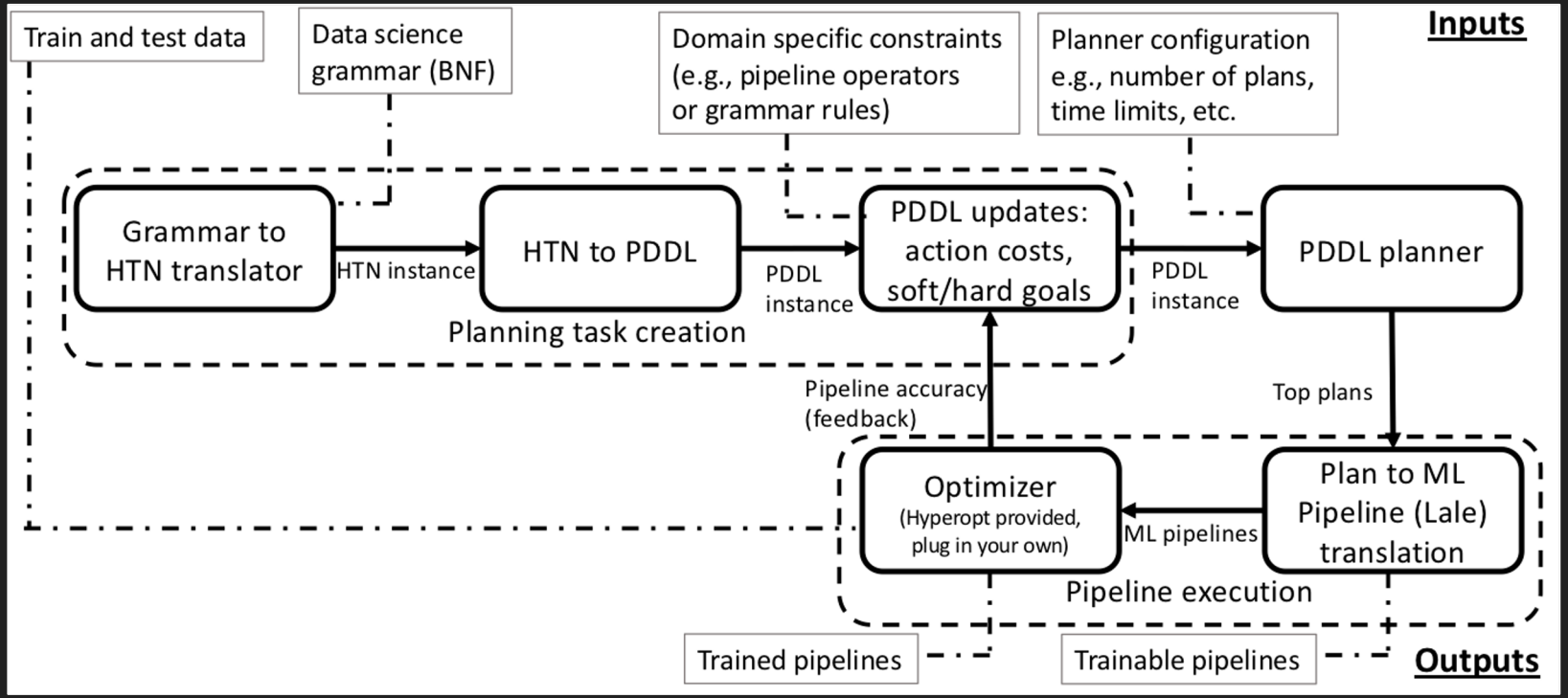
– Use regular grammars to define the possible pipeline compositions

– Translate regular grammars to HTN planning model, and then to classical planning model

– Enrich planning model with user defined constraints

– Use AI Planners to solve the planning model and translate plans to pipelines

ML pipeline generation/exploration walkthrough



ML pipeline exploration

Do it yourself: https://github.com/IBM/grammar2pddl

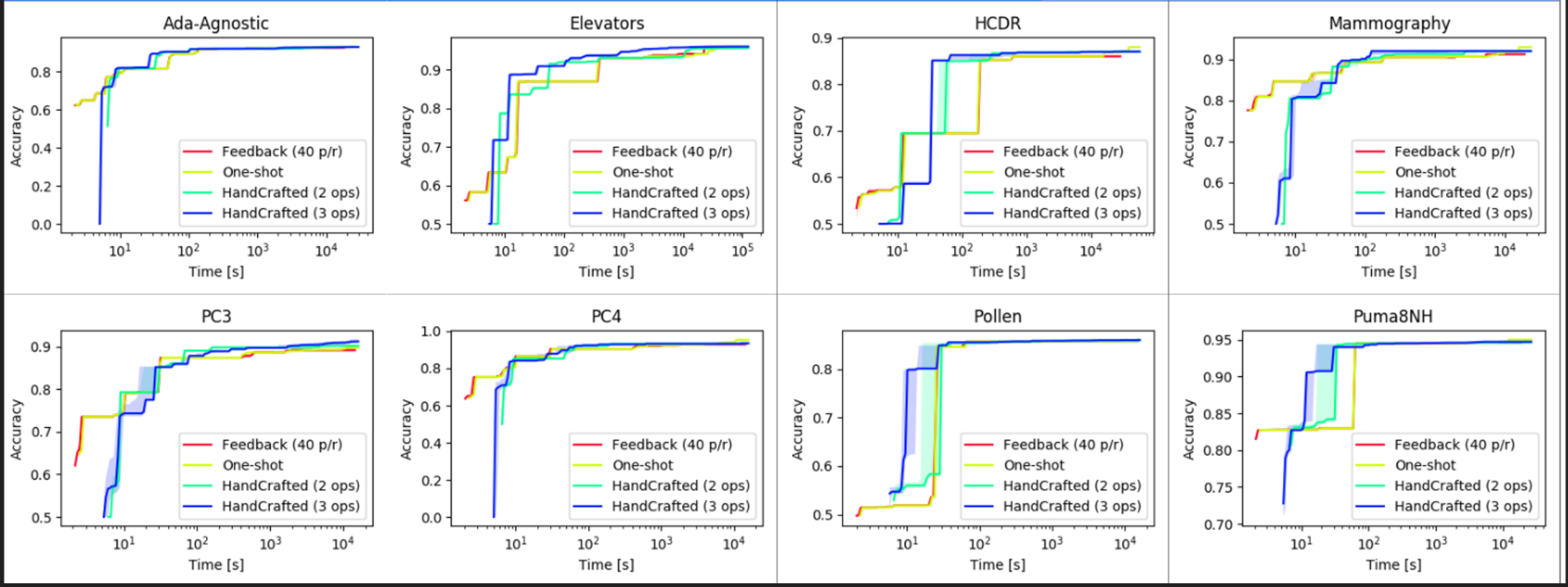
Tools and links for hands-on exercise

- HTN2PDDL translation: https://github.com/ronwalf/HTN-Translation
- K* planner: https://github.com/ctpelok77/kstar
- PlanUtils: https://github.com/AI-Planning/planutils
- PDDLEditor plugin for VSCode: https://marketplace.visualstudio.com/items?itemName=jan-dolejsi.pddl

ML pipeline exploration: matching handcrafted pipeline performance

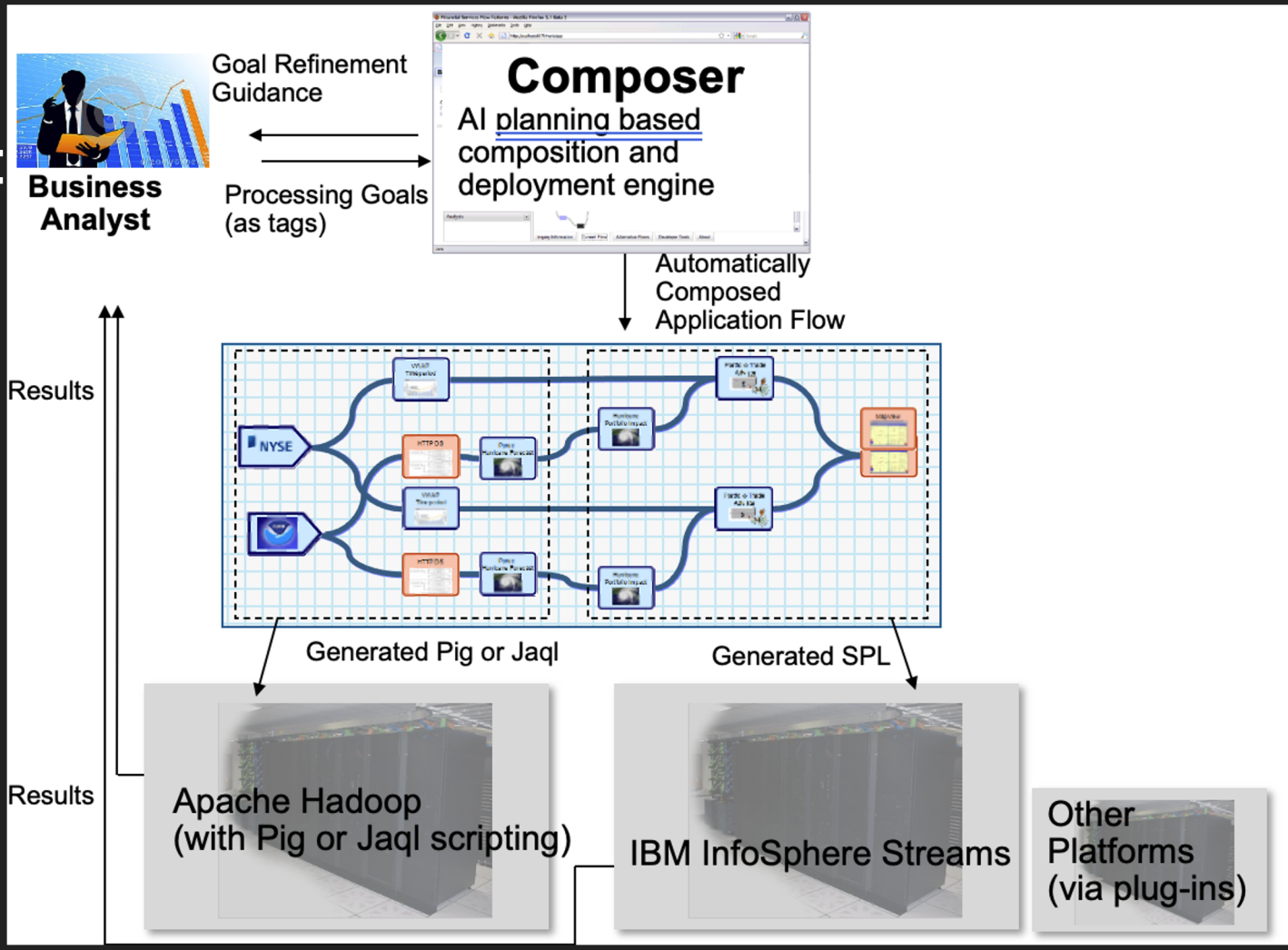
- **27 datasets from OpenML (shown datasets with max accuracy ≤ 95%)**
- **Two hand-crafted baselines: manually chosen by experts chains of 2 or 3**

- **W.o. feedback: one-shot 1000 plans**
- **With feedback: 40 plans per iteration**

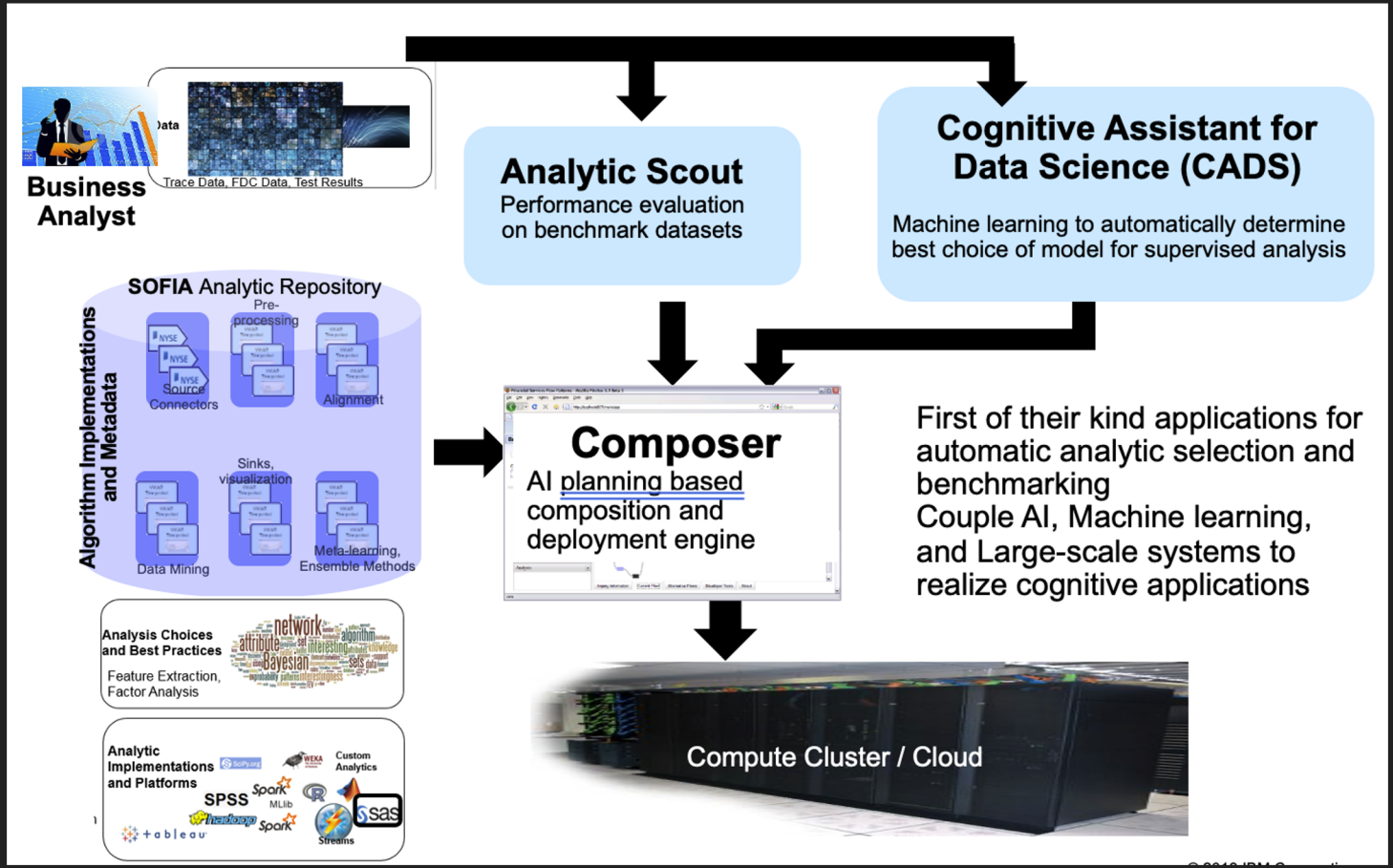


Other Applications:

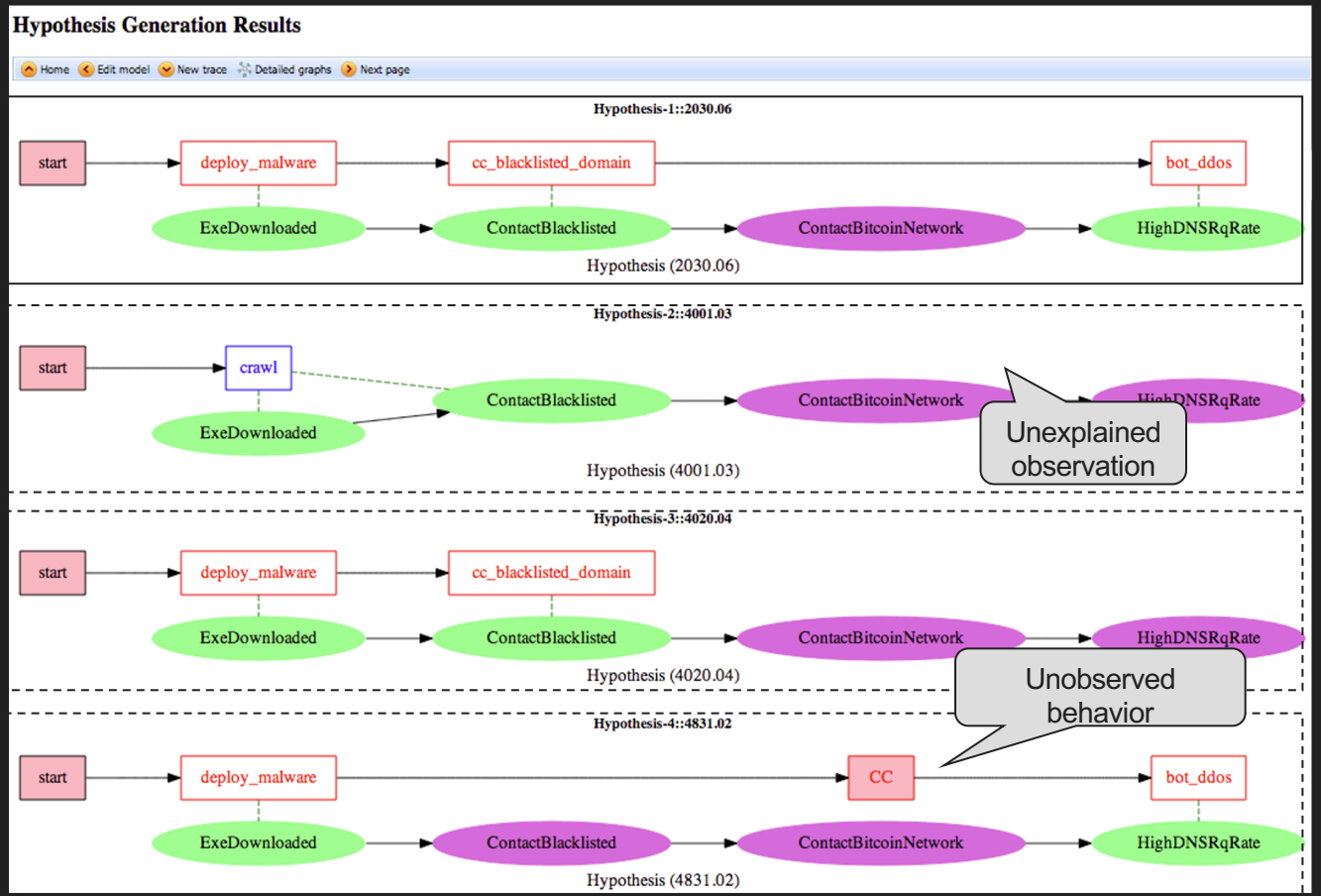
IBM Automated Analytics Composer



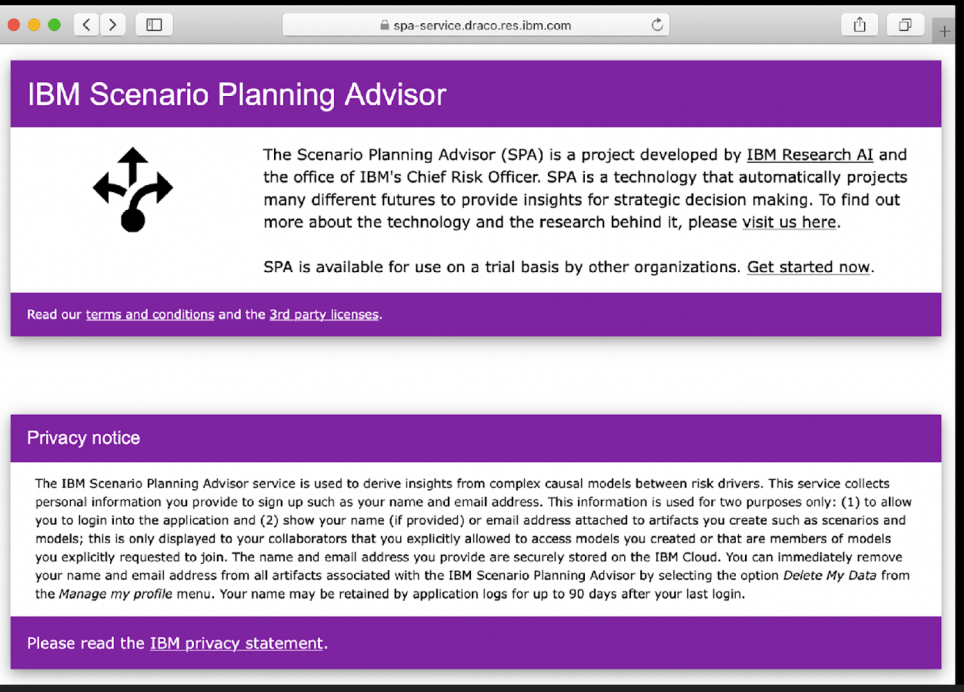
Other Applications: DS automation via Scout and CADS



Other Applications: Hypothesis generation in cyber



Scenario Planning Advisor



Problem

Scenario Planning for risk mitigation is a mostly manual process

Only a few scenarios can be constructed manually and explored

High impact low likelihood events are overlooked

Benefits

- Reduction in time for building scenarios from months to hours
- Exploration of orders of magnitude more scenarios than possible if built manually

Solution

– Exploit NLU techniques to semi-automatically construct scenario planning models

– Automatically explore the space of possible scenarios with an AI Planner

– Choose scenarios of high relevance to a client at a particular time