Data & AI Hackathon

Feb 17-20, 2026

Challenge #1: Job Description Clustering & Standardization

Section	**Description**
Title	**Problem 1 – Job Description Clustering & Standardization**
Business context	Methanex has ~2,000 job description profiles, close to a 1:1 ratio with employees. This makes it difficult to: ● Standardize positions, titles, and career paths ● Align HR processes (recruitment, compensation, performance management) ● Identify job families and overlapping roles across functions and locations **We need a data-driven view** of how similar or different current job descriptions are to move toward a standardized set of roles.
Problem statement	Given ~2,000 job descriptions (plus optional metadata), the goal is to: ● **Cluster similar job descriptions** into groups ("job families") ● Understand **how close** job descriptions are to each other ● Identify **what makes clusters different** (responsibilities, skills, departments, etc.) Key questions: How many natural job families exist? Which roles are very similar and could be standardized? Which roles are truly unique?

Expected solution – modeling	Teams are expected to: 1. **Represent job descriptions:** Use NLP techniques (e.g., TF-IDF, word/sentence embeddings) to convert text into numerical vectors. 2. **Cluster job descriptions:** Apply clustering algorithms (e.g., k-means, hierarchical, DBSCAN, etc.). Explore different numbers of clusters/parameters and justify the choice. 3. **Interpret and label clusters** • Identify what makes each cluster coherent (shared responsibilities, skills, departments). 4. **Compare clusters** • Highlight what makes clusters different from each other (keywords, skill sets, typical job levels).
Minimum Deliverables	1. **Cluster visualization** – 2D plot (e.g., PCA/t-SNE/UMAP) showing each job description as a point, colored by cluster. 2. **Cluster overview** – For each cluster: a short label (e.g., “Maintenance”, “Finance”), 3–5 key keywords, and 2–3 example job titles. 3. **Standardization insights** – be able to provide answers to questions such as: “Which clusters could share a common job profile?” and “Where do we see truly unique roles (outliers)?, and more

Challenge #2: Safety Incident & Near-Miss Pattern Mining

Section	**Description**
Title	**Problem 2 – Safety Incident & Near-Miss Pattern Mining**
Business context	Methanex records **incidents, near misses, and lessons learned** (e.g., root cause analyses, hazard categories, and corrective actions). This information is valuable, but it's often reviewed case by case. There is an opportunity to use data science to **spot recurring patterns**, understand **drivers of higher-severity events**, and **inform prevention strategies** (training focus, procedures, safeguards).
Problem statement	Given historical safety-related records (incidents, near misses, or RC cases), the goal is to: ● Identify **patterns and clusters** of similar events (e.g., by activity, equipment, cause). ● Understand which factors are most associated with **higher severity** or **high potential** events. ● Provide **simple, data-driven recommendations** on where to focus prevention efforts.
Expected solution – analysis & modeling	Teams are expected to: ● **Explore the dataset:** Show basic statistics: counts over time, by site, by event type, by severity. 2. ● **Find patterns and clusters:** Use structured fields and/or text (via keywords/embeddings) to group similar events. ● Identify typical "scenarios" (e.g., maintenance-related, loading/unloading, office-based). ● **Analyze severity drivers:** Explore which conditions or categories are more often linked to higher severity/potential. ● Optionally build a simple model to predict **high vs. low severity** based on event characteristics.

Minimum Deliverables	**Pattern & trend visuals** – 1–3 clear charts (e.g., bar charts or time trends) showing key patterns (by activity, site, hazard category, etc.). **Event clusters or scenarios** – A simple summary of 3–6 “typical event groups” (e.g., “maintenance valve work”, “forklift operations”) with a few keywords/phrases for each. **Prevention insights** – Which types of events or situations deserve the most attention for prevention, based on the data?

Challenge #3: Early Detection of Process Excursions from Sensor Data

Section	**Description**
Title	**Problem 3 – Early Detection of Process Excursions from Sensor Data**
Business context	Methanex operates large continuous process plants with many critical assets (e.g., reformers, compressors, distillation columns). These assets generate thousands of **time-series tags** (sensor signals) stored in a historian. When something drifts out of its normal operating range, it can lead to **unplanned downtime, quality issues, or safety risks** if not caught early. Today, many deviations are detected **reactively** (alarms, operator experience). Methanex would like to explore **data-driven methods** to detect **early signs of abnormal behavior** so operators and engineers can act before a major excursion or trip occurs.
Problem statement	Given historical time-series data from one or more critical process tags (and possibly related tags), the goal is to: ● **Learn what "normal" behavior looks like** for a signal (or small set of signals). ● **Detect and flag anomalies or excursions** where the behavior deviates from normal. ● Highlight **when** issues start and how far in advance they could have been detected.
Expected solution – modeling	Teams are expected to: ● **Explore & clean the data:** Handle missing values, obvious outliers, and resampling if needed. ● **Model normal behavior** • Use time-series / anomaly detection techniques (e.g., moving statistics, isolation forest, autoencoder, forecasting plus residuals) to learn "normal" patterns. ● **Detect anomalies / excursions** • Produce an anomaly score or classification over time. • Compare detected anomalies with any known events (if provided).

Minimum Deliverables	• **Anomaly timeline visualization** – Time-series plot(s) showing the main tag(s) over time with anomalies clearly marked (e.g., highlighted or flagged). • **Simple detection logic / model** – A short description of the method (e.g., “we trained a forecasting model and flagged points where the error exceeded X”, or “we used isolation forest on sliding windows”). • **Operational insights** – In which periods do we see abnormal behavior? or Could some events have been detected **earlier** than they actually occurred?