Patryk Seweryn

DATA SCIENTIST & AI ENGINEER

I help businesses turn complex data into revenue, manual processes into automated workflows, and ambitious AI ideas into systems that actually ship.

Over 8 years I've helped companies across industries solve hard problems with AI — reducing false alarm rates in security systems, making millions of products searchable in natural language, automating regulatory compliance that used to take teams of analysts.

Curious how AI could work for your business? Let's talk — I'd love to hear about your challenge.

---

linkedin.com/in/patryk-seweryn
patryk.seweryn@gmail.com

Contents

LLMs & AI Agents

AI Platform for Regulatory Screening

Automating regulatory screening so compliance teams never miss a flag.

Banks must screen every transaction against regulatory watchlists — fragmented across authorities, constantly updated, arriving in every document format. I built a platform that **extracts entities from incoming documents, matches them against consolidated watchlists**, and produces full audit trails.

PROBLEM

Compliance analysts **manually cross-referenced documents** against fragmented watchlists from multiple authorities — slow, error-prone, with serious consequences for misses.

IMPACT

Replaced manual cross-referencing with **automated, auditable screening** — reliable coverage with a clear paper trail for regulators.

Python LLMs pydantic-ai Streamlit Docker FastAPI

LLM-Powered E-commerce Search Engine

Making products findable through natural language search.

Customers of a large e-commerce platform struggled to find what they needed — especially when they didn't know the exact product name. I built a search engine that **understands everyday language** and returns the right results, even in grammatically complex Polish. Then took it further by iterating on accuracy in a second phase.

PROBLEM

Traditional keyword search **failed for shoppers who described needs in plain language** rather than exact product names.

IMPACT

Made the shopping experience **accessible to non-technical users** by bridging the gap between how people talk and how products are categorized.

Python LLMs RAG LangChain FAISS

Search Intent Classifier for E-commerce

Understanding what users actually want when they search.

Search queries rarely say exactly what people mean. I built a classification system that **identifies the intent behind search queries** — what the user is actually trying to accomplish — using an ensemble of text embeddings and machine learning.

PROBLEM

Raw search queries are ambiguous — **the same words can mean different things** depending on context, making it hard to serve relevant results.

IMPACT

Proved that **user intent can be reliably extracted** from search queries, laying groundwork for smarter search behavior.

Python · XGBoost · UMAP · Sentence Transformers · FastAPI · Docker

---

LLM-Powered Customer Feedback Explorer

Turning customer feedback into answers you can just ask for.

Customer feedback comes in piles of documents — surveys, spreadsheets, reports. I built a system that extracts questions and answers from these files and lets users **query the results conversationally**, like asking a colleague who's read everything.

PROBLEM

Analyzing customer feedback meant **manually reading through stacks of documents** in various formats — slow and hard to get a big-picture view.

IMPACT

Turned **hours of manual document review into an interactive conversation** with the data.

Python · LLMs · Prompt Engineering

Automated Document Router for Business Workflows

Sorting business documents automatically so nobody has to read them first.

Contracts, invoices, forms — every document needed to land in the right category, and someone had to read it to decide where. I built a system that **classifies documents automatically based on their content**, removing the need for manual review.

PROBLEM

Documents had to be **manually read and sorted** into categories — tedious, slow, and error-prone at scale.

IMPACT

Eliminated manual document sorting, letting documents route themselves to the right category instantly.

Python DVC Poetry Tesseract OCR SciPy

Customs Tariff Classifier

Assigning customs codes automatically so humans don't have to look them up.

Every product that crosses a border needs a customs tariff code — and someone has to assign it. I built a classifier that **learns from previously coded products to automatically assign tariff codes** to new ones.

PROBLEM

Assigning customs tariff codes to products was a **manual lookup process** — slow, requires specialist knowledge, and doesn't scale with growing catalogs.

IMPACT

Automated tariff code assignment, removing a manual bottleneck from the import/export workflow.

Python Scikit-Learn Flask

Automated Product Recommendations for E-commerce

Recommending products by what they actually are, not what others bought.

Most recommendation engines rely on purchase history — but that doesn't help when you want products with similar specs. I built a system that **matches products by their technical features**, replacing a manual process where people paired 'similar' items by hand.

PROBLEM

Product recommendations were either based on purchase patterns (irrelevant for feature comparison) or **manually curated — expensive and impossible to maintain at scale**.

IMPACT

Automated product pairing across the entire catalog, eliminating a labor-intensive manual process while delivering more relevant recommendations.

Python Decision Trees Dash

Data & Analytics

Proactive Failure Detection for Smart Thermostats

Detecting thermostat failures before customers notice.

Instead of waiting for customers to report a broken thermostat, I built an analytics framework that **detects failures proactively** — analyzing temperature and operational data to catch problems early and classify what went wrong.

PROBLEM

Thermostat failures were only discovered **when customers complained**, leading to slow response times and poor user experience.

IMPACT

Enabled **proactive failure detection**, letting the company reach out to customers before they even noticed something was wrong.

PySpark · Databricks · Delta Tables · Azure

ML-Driven Efficiency Optimization for Gas Water Heaters

Squeezing more efficiency out of home water heaters through data.

Home water heaters waste energy when they run suboptimally — but without data, it's hard to know what 'optimal' even looks like. I analyzed operational data and built ML models to understand **how to get the most efficient performance** out of gas water heaters.

PROBLEM

Gas water heaters often ran inefficiently, but there was **no clear understanding of what optimal operation looks like** for different usage patterns.

IMPACT

Provided **data-driven insights into optimal heater operation**, enabling more efficient energy use in residential settings.

Pandas · NumPy · Scikit-Learn · PySpark

Revenue Prediction by Location

Predicting store revenue before opening the doors.

Where should the next store go? I built predictive models that **estimate revenue for potential locations** based on geospatial data, plus an interactive dashboard for exploring predictions and comparing sites — turning expansion planning from gut feel into data-driven decisions.

PROBLEM

Choosing new store locations was based on **intuition and manual research**, with no reliable way to predict how a location would perform before committing.

IMPACT

Gave expansion teams a **data-driven tool for evaluating potential locations**, reducing the guesswork in site selection.

Python XGBoost Scikit-Learn SHAP PostgreSQL Dash Plotly Leaflet OpenStreetMap

Computer Vision

Smart Delivery Detection for Home Cameras

Spotting package deliveries from home camera feeds.

Home cameras generate hours of footage, but owners only care about specific moments — like when a package shows up. I built a detection system that **identifies deliveries from camera feeds**, backed by a full cloud pipeline for training and deploying the model.

PROBLEM

Homeowners had **no automated way to know when a package arrived** without manually checking camera footage.

IMPACT

Gave users **instant delivery notifications** from their existing home cameras.

Python PyTorch ONNX OpenCV Azure Cloud

DPV Detector (Drones / People / Vehicles)

Real-time detection of drones, people, and vehicles on edge hardware.

Security systems need to detect threats in real time, on-site, without relying on cloud infrastructure. I built an object detector that identifies drones, people, and vehicles from both thermal and RGB camera feeds, **running directly on edge hardware at ~15 FPS**.

PROBLEM

Existing detection systems either **required cloud connectivity or couldn't handle multiple sensor types** in real time on lightweight hardware.

IMPACT

Delivered a working real-time detector used in **live demonstrations for prospective clients** — directly supporting sales efforts.

Python NVIDIA Jetson Detectron2

Training Data Pipeline for Detection Models

Building better training data through systematic curation.

A model is only as good as its training data. I built a pipeline that **systematically evaluates and filters video datasets** to select the most effective training samples — not by hand, but through statistical analysis.

PROBLEM

Training data was selected **manually or arbitrarily**, with no systematic way to know which samples actually improve model performance.

IMPACT

Turned dataset curation from guesswork into a **repeatable, data-driven process** that directly improved detection model accuracy.

Python PyTorch Pandas NumPy Matplotlib

Deep Learning Denoiser for Thermal Cameras

Making thermal camera footage actually usable.

Thermal cameras produce inherently noisy images — artifacts and distortions that make the footage hard to use for anything downstream. I applied deep learning denoising techniques to clean up infrared imagery, **significantly reducing noise across multiple frequency bands**.

PROBLEM

Raw infrared camera output suffered from **heavy noise and artifacts**, degrading image quality to the point where downstream processing was unreliable.

IMPACT

Dramatically improved thermal image quality, making the footage usable for further analysis and processing.

TensorFlow

Fine-tuning Counter-UAV Detection Model

Cutting false alarms in drone detection without losing real threats.

A counter-UAV detection system was triggering too many false alarms — especially confusing birds with drones. I **fine-tuned the model to reduce false positives** while keeping detection reliable across all target categories.

PROBLEM

The existing model **flagged too many non-threats as drones**, overwhelming operators with false alarms and undermining trust in the system.

IMPACT

Reduced false alarm rates while maintaining detection performance, making the system practical for real-world deployment.

Python Computer Vision

3D Body Scan Denoiser for Mobile LiDAR

Cleaning up noisy 3D body scans from phone cameras.

Phone LiDAR sensors can scan a person's body in 3D, but the raw data is noisy — full of artifacts that make measurements unreliable. I built a deep learning model that **cleans up these scans**, trained on synthetic 3D data to achieve reliable accuracy on real scans.

PROBLEM

Raw 3D scans from iPhone LiDAR were **too noisy for reliable body measurements**, with artifacts that distorted the actual shape.

IMPACT

Made phone-based 3D body scanning **accurate enough for practical use** by removing noise without losing body shape detail.

Python PyTorch Open3D Blender

Semantic Label Clustering for Large-Scale Detection

Making sense of model predictions when there are hundreds of labels.

When an object detector has hundreds of classes, analyzing its predictions becomes overwhelming. I built a tool that **semantically groups similar labels** — so instead of reviewing hundreds of classes one by one, analysts can focus on meaningful clusters.

PROBLEM

Models with large label sets produced predictions that were **nearly impossible to analyze at a glance** — too many classes, too many confusion patterns.

IMPACT

Made model analysis practical for large label sets by letting analysts **work with intuitive groups** instead of individual classes.

NLP Computer Vision