Case Study: Predicting Corporate Cash Burn Using AutoML Tools

Author: Alex Havryleshko\ **GitHub Repo**: https://github.com/havryleshko/10k_reports\ **Tech Stack**: Python, Pandas, PyCaret, H2O.ai, Matplotlib

---

Problem Overview

Cash burn is a signal for startups and capital-intensive companies. My goal was to predict whether a public company will "burn cash" (i.e., have negative free cash flow) in the next fiscal year, using historical financial statement data. I have taken data fof 32 public companies in Renewable Energy sector between 2021 - 2024 from Perplexity Finance

---

Dataset

- **Source**: Manually extracted annual data from public 10-K reports and financial database (https://www.perplexity.ai/finance)
- **Scope**: 32 companies (renewable/nuclear energy and industrial sectors), 2021–2024.
- **Size**: 123 company-year rows after cleaning (melted down)

Target: `burn_cash` (1 if FCF < 0 next year, else 0). Binary classification

---

Workflow Summary

1. Data Preparation

- Combined balance sheet and cash flow statements (separate file)
- Added future cash flow targets (shifted FCF & OCF)
- Handled missing data and ensured temporal alignment

2. Feature Engineering

- Derived \~10 key financial ratios: profitability, investment, working capital change
- Calculated YoY deltas for Net Income, FCF, and OCF
- Created binary `burn_cash` label based on next-year FCF

3. AutoML Modeling

- **PyCaret** (main pipeline): classification task with 10-fold cross-validation (used it eventually as the dataset is too small for h2o)
- **H2O.ai** (experimental): used H2OAutoML with leaderboard
- Evaluated models on metrics: Accuracy, AUC, F1, MCC, Recall

---

Results

Top PyCaret models:

Model	Accuracy	AUC	F1 Score	MCC
Decision Tree	0.93	0.95	0.93	0.90
AdaBoost	0.93	0.95	0.93	0.90
Gradient Boosting	0.93	0.95	0.93	0.90

Despite high metrics, I observed potentially some signs of **overfitting**:

- Dataset is small (123 rows), even after 32 companies' data
- Performance varies significantly between folds

Challenges & Lessons Learned

- **Data size limits generalization**: even with clean preprocessing, real-world modeling is bound by the size and diversity of input data
- **AutoML is not a magic wand**: both PyCaret and H2O.ai need thoughtful feature engineering and interpretability checks
- **Iterative design matters**: I revised features, tried multiple targets, and evaluated both traditional and AutoML tools

---

What I Would Do Next

- Expand the dataset to 300+ companies across 5–10 years
- Use quarterly data
- Incorporate NLP from 10-K text sections (e.g., risk factors)
- Build a dashboard where businesses can input financials to assess burn risk (deployment issue)

Final Thoughts

This project trained me to catch signal through financial noise and taught me that the **hardest part of ML is data curation.**

View the code on GitHub: https://github.com/havryleshko/10k_reports