Fraud Detection: An Interactive Dashboard Solution

IBM Hackathon 19 - Finance Track

A concise overview of our solution for detecting fraudulent financial transactions, leveraging machine learning and an interactive dashboard for rapid analysis.

Tax Fraud in France: A Critical Challenge

Tax fraud is no small game — it costs France tens of billions of euros every year!

That’s money that could be funding schools, hospitals, and smoother roads (or at least fewer potholes).

And when the state’s wallet feels light, pension plans start to sweat — meaning we might all work a little longer than we’d hoped.

The solution? More smart controls, less creative accounting, and a strong dose of tax fair play.

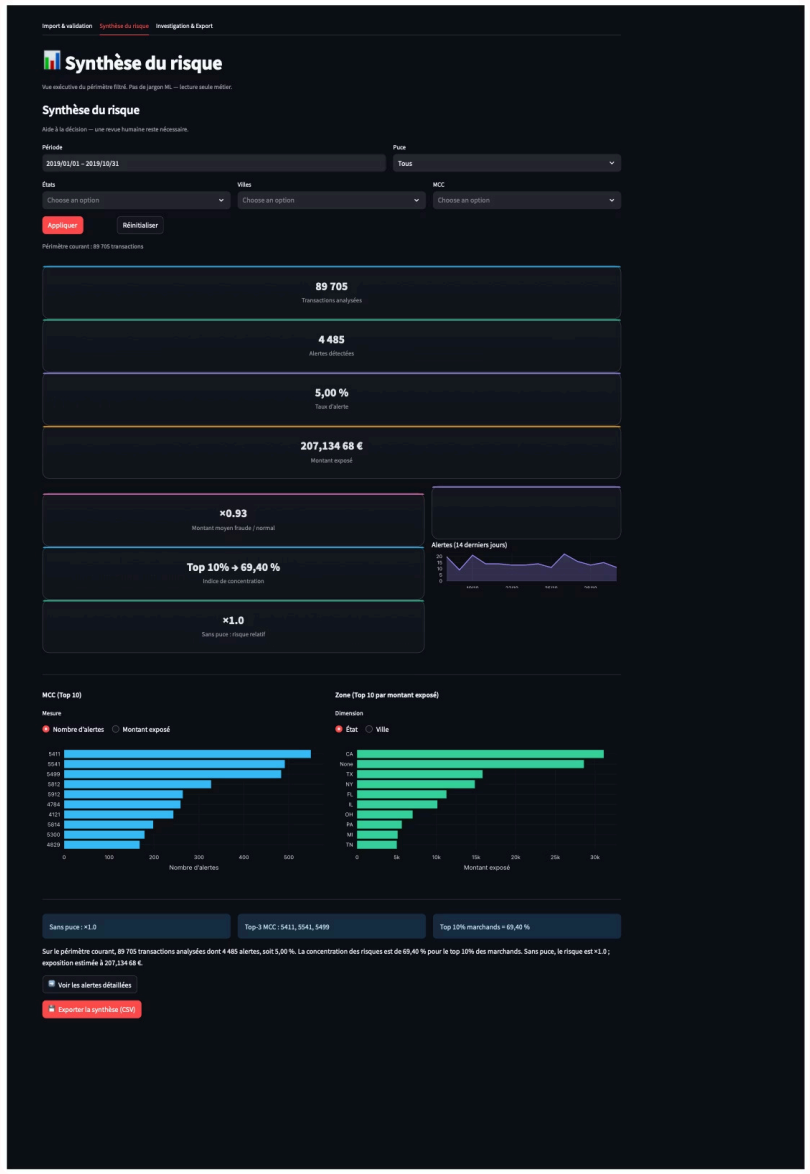
The Solution: Interactive Fraud Prediction Dashboard

Our Goal: Instantaneous Detection

We developed a powerful, yet user-friendly, dashboard aimed at streamlining the detection of fraudulent transactions across various datasets. Our primary objective was to provide finance professionals with a clear, immediate view of risk.

Key Features:

- Interactive user interface for easy data integration.
- Real-time prediction of fraudulent and suspicious activity.
- Comprehensive summary and visualization dashboard.



How It Works: Seamless Prediction Workflow

The core of our solution is a streamlined process that converts raw data into actionable insights, requiring minimal user intervention.

Data Ingestion

The user drags and drops a raw dataset (CSV file) into the interface.

ML Prediction

A pre-trained machine learning model processes the data to identify anomalies.

Flagging Transactions

The model flags transactions as fraudulent or highly suspicious based on risk scores.

Recap Dashboard

The results are displayed in an intuitive dashboard for immediate review and action.

Chapter 1: Exploratory Data Analysis (EDA)

Our initial phase involved rigorous data exploration to ensure the integrity and quality of the input data before any modeling could begin.

Missing Values Check

Systematic identification and appropriate treatment of columns containing missing data to prevent bias.

Column Formatting

Standardization of data types and column names for consistent and efficient downstream analysis.

Descriptive Statistics

Calculation of key metrics (mean, median, standard deviation) to understand the distribution and variance of variables.

Visualizing the Data & Final Preparation

Key Visualizations

We used visual representations like scatter plots and histograms to uncover critical trends, anomalies, and underlying relationships between different variables.

- Identify hidden patterns related to fraudulent activity.
- Validate assumptions about variable distributions.
- Detect outliers that require special handling.

The Merged Dataframe

A crucial step was merging all input data files into a single, cohesive dataframe. This complete and consistent dataset served as the foundation for our most complex modeling efforts.

The merged dataframe ensured that every piece of information was available for deep-dive analysis and full-feature model training.

Chapter 2: Advanced Analysis and Transformations

Moving beyond cleaning, we performed sophisticated data manipulations to prepare the features for machine learning algorithms.

Data Preparation

Final cleaning and organization steps to structure the data for optimal model ingestion.

Categorical Encoding

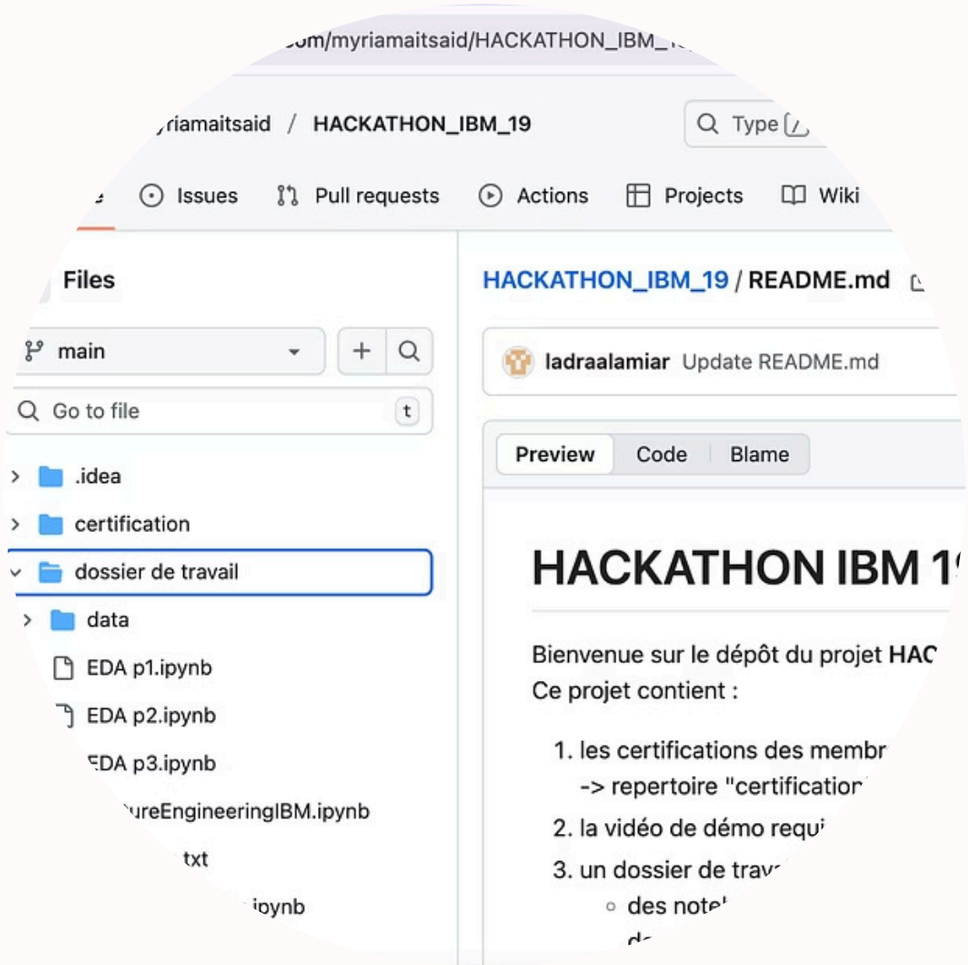
Conversion of categorical features (e.g., location, transaction type) into numerical variables that models can process effectively.

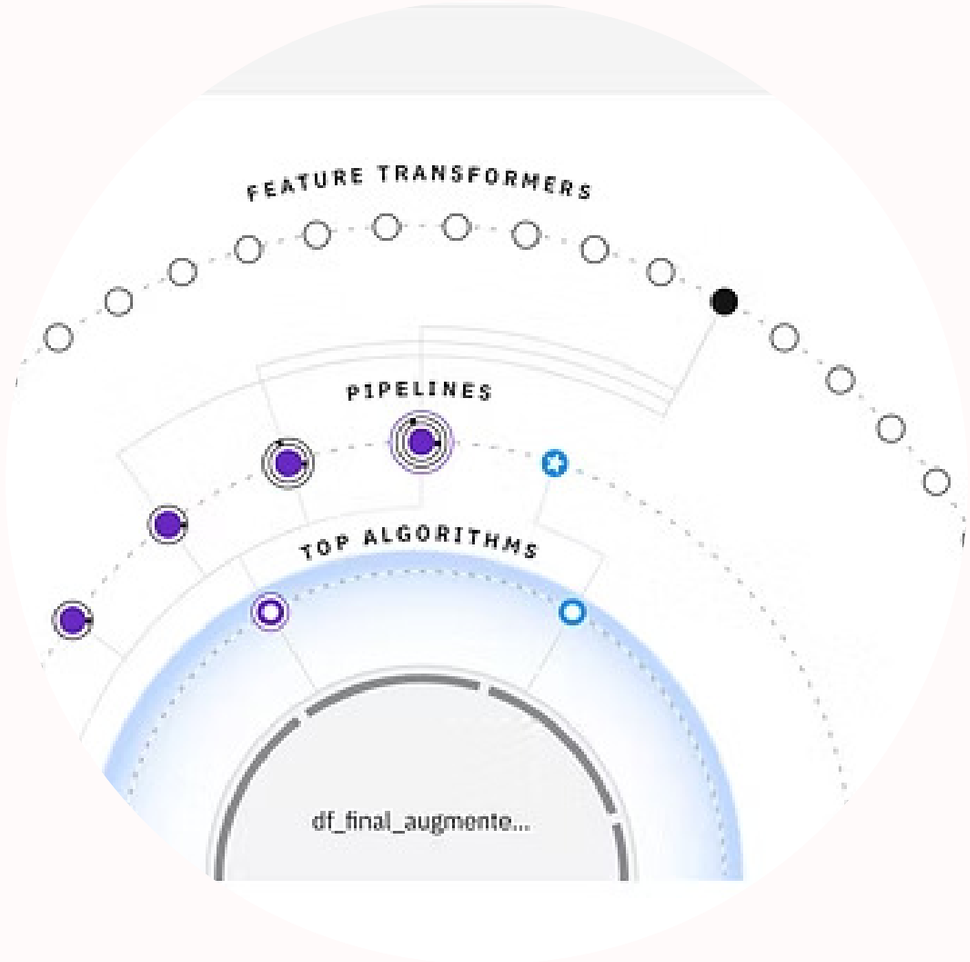
Correlation & PCA

Analyzed variable relationships and applied Principal Component Analysis (PCA) for dimension reduction, guiding model selection and improving efficiency.

Chapter 3: Modeling and Conclusions

Our modeling approach involved a hybrid strategy, utilizing both familiar environments and powerful IBM cloud tools to maximize efficiency and performance.





In-House Modeling (Python)

A significant portion of the modeling was executed in Python, extending from the EDA phase to provide granular control over algorithms.

IBM Watsonx Platform

We used the Watsonx platform to accelerate workflow, facilitate rapid model deployment, and benchmark performance against our Python models.

Strategic Model Comparison

The hackathon provided an opportunity to compare model efficiency and discover how IBM Watsonx can simplify the machine learning pipeline.

Workflow Acceleration

Understanding how a platform like Watsonx can significantly cut down the time required for model training, testing, and tuning.

Performance Benchmarking

Directly comparing the performance metrics of models developed in Python versus those optimized within the IBM platform.

Efficiency Assessment

Evaluating the overall gain in time and efficiency delivered by adopting an integrated ML platform.

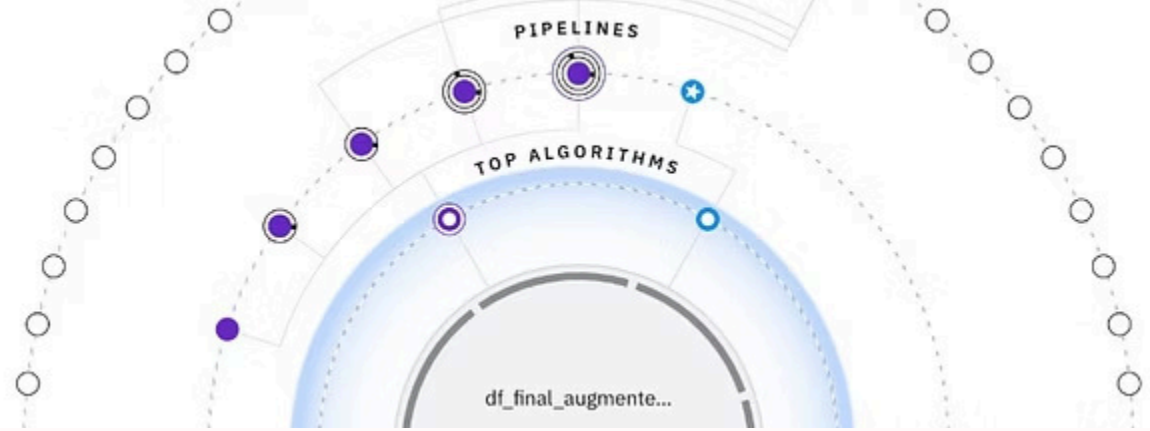
Technical Challenges & Future Improvements

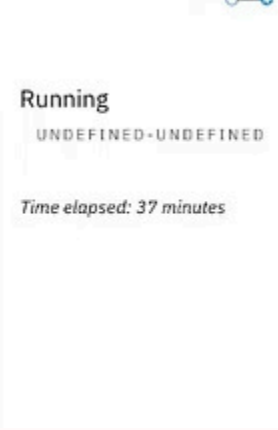
Challenges Encountered

- Watsonx credit limitations forcing multiple project recreations
- Column mismatch between merged training data and evaluation dataframes
- Steep learning curve for platform adoption

Proposed Improvements

- Enhance model performance metrics
- Develop robust feature handling for missing columns without compromising accuracy
- Enable predictions on incomplete dataframes





Testing Hypothesis: Data Simplicity vs. Performance

We tested two distinct datasets to determine the optimal balance between data complexity and model performance for a practical demonstration.

1. Full Merged Dataframe

This set included all available columns, providing the maximum feature set for potentially achieving the highest predictive accuracy.

2. Pre-Merge Dataframe

Used to test if a robust model could be built without the supplementary columns, aiming for a simpler, faster-to-process dataset.

This comparison allowed us to determine if a near-identical performance could be achieved with a reduced feature set, making the final model more efficient for deployment.

Next Steps: Selection and Validation

The final stage of our project focused on selecting the best model candidate based on rigorous performance evaluation.

1

Model Selection

Identification of the most accurate and efficient algorithms for fraud prediction based on the comparative analysis results.

2

Performance Evaluation

Measuring key metrics (e.g., precision, recall, F1-score) to confirm the robustness of the chosen models.

3

Final Candidate

Determining the optimal model for production or demonstration that maximizes accuracy while maintaining efficient processing times.

Our solution provides a reliable, high-performing framework for real-time financial fraud detection.