INTELLIGENT ANALYTICS HACKATHON - 2026

Team UM6P

Team Members: Ibrahima FARO [ibrahima.faro@um6p.ma]
Aya ALAMI [aya.alami@um6p.ma]
Mariam DIAKITE [mariam.diakite@um6p.ma]
Babacar Sanding [babacar.sanding@um6p.ma]

Team Referents: El Gargouh Younes & Nadif Firdaouss

Customer Profiling and Risk Management in Automotive Insurance

January 19, 2026

Contents

1 Context

This project, developed by Team UM6P as part of the DXC "Intelligent Analytics" Hackathon, aligns with a global movement of organizational transformation through data and artificial intelligence. In many sectors, decision-makers must manage growing volumes of data while facing increased demands for speed, traceability, and efficiency in decision-making. However, traditional approaches often remain descriptive and reactive, producing after-the-fact analyses that heavily rely on technical teams. Intelligent analytics, on the contrary, aims to automate data preparation, accelerate access to information (including through natural language interfaces), and enhance decisions using predictive models and actionable recommendations. This dynamic is particularly important in activities where uncertainty, variability in behaviors, and economic trade-offs require proactive risk and performance management.

2 Justification for Choosing the Automotive Sector

In the Morocco Road Safety Report by the National Road Safety Observatory [1], the choice of the automotive insurance sector is particularly solid and interesting for 3 reasons: impact, volume of data, and immediate business value.

The report highlights that Morocco recorded 113,625 injury accidents in 2022, with 3,499 fatalities. This mortality level exceeded the interim target of the national strategy (target $<$ 2,643 deaths in 2022). This result shows that the automotive sector presents a frequent and costly risk, perfectly aligned with an "Intelligent Analytics" AI solution focused on prevention - pricing - steering, and especially in proposing recommendations.

The report emphasizes that reliable data is a key lever to: understand causes, target risk factors, evaluate the impact of interventions, and make evidence-based decisions. This is exactly what we intend to do in this project: *transform contract histories into indicators (premium/day, loss ratio, customer risk, fraud...) and then into decisions.*

From an economic weight perspective, the report recalls that road accidents represent an estimated economic burden of about 3% of the GNP of countries and the passenger car fleet is very significant (44.7% of the national fleet, and 50.91% of vehicles involved in injury accidents are passenger cars). Thus, the auto market offers many contracts, many exposures, therefore a lot of data and potential gains (better segmentation, better premium, better prevention, better claims management).

In Africa, these issues are even more critical. The continent has the highest road death rate in the world, with about 19.6 deaths per 100,000 inhabitants, compared to a global average of about 15 per 100,000 (WHO), even though it represents only about 3% of the global vehicle fleet. Paradoxically, Africa is experiencing rapid motorization growth, driven by urbanization, the rise of the middle class, and the development of transport activities (World Bank, Africa Transport Outlook). This dynamic mechanically increases exposure to automotive risk, in contexts where infrastructure, control systems, and insurance mechanisms often remain insufficient.

We chose automotive insurance because it is a universal risk, with high frequency and highly variable costs, making it an ideal use case for AI. Globally, insurers seek to improve technical profitability (loss ratio), accelerate claims management, and better manage customer retention in a context of competition and evolving costs. In Africa, urban growth and the intensification of uses (fleets, mobility) make risk exposure more complex, while data quality can be heterogeneous: an intelligent analytics solution that cleans, structures, and transforms data into actionable indicators (risk, fraud, payment, termination) brings immediate gain for business decisions.

3 Global Objective

Automotive insurance today constitutes a strategic but highly risk-exposed segment for insurers. The increase in claims, the evolution of policyholder behaviors, and intensifying competition make portfolio management more complex, while traditional approaches often remain descriptive and reactive. This project therefore aims to mobilize AI and advanced data analysis to transform the identification and management of at-risk customers, shifting from an a posteriori logic to a predictive, proactive, and business decision-oriented approach.

4 Central Problem

Thus, to operationalize this objective and precisely frame the project's contribution, we formulate the central problem as follows:

How can we effectively anticipate and manage at-risk customers in automotive insurance to reduce financial losses and improve business decision quality through artificial intelligence?

This problem is at the heart of current issues in the insurance sector, where performance increasingly depends on the ability to anticipate future policyholder behaviors rather than simply observing past events. This problem is set in a context marked by:

- increasing claims frequency, linked to the growth of the vehicle fleet, urban density, and evolving driving behaviors, exerting direct pressure on insurer profitability.
- high contract termination rates, particularly in automotive insurance, where customers are increasingly volatile and price-sensitive, making retention complex and costly.
- strong information asymmetry between insurer and policyholder, especially regarding actual driving behaviors, latent risks, or termination intentions, complicating fine risk assessment.

- and decisions still largely reactive rather than predictive, based on fixed rules or descriptive analyses, which often occur after claims arise or customer loss, instead of acting proactively.

5 Ambition

The ambition driving us is to strengthen the decision-making capacity of automotive insurance stakeholders, by providing them with tools capable of detecting weak signals, anticipating risks, and guiding pricing, retention, and prevention strategies.

The implementation of this application could allow us to present the product to insurers while highlighting the additions we made compared to the classic method.

We thus aspire to make available to users an easy, simple, automated, and intelligence-based interface, specifically trained in the domain.

6 Training Data

The project relies on a dataset from an automotive insurance portfolio, containing 26,383 rows corresponding to contract records.

Originally, the dataset included **12 main variables**, describing essentially:

- contract and invoice identification (e.g., *num_contrat*, *Num_facture*);
- temporal information (*datedeb*, *datefin*, *datcpt*, *exe*);
- premium amounts (*Prime*);
- operation nature (*libop*);
- contract renewal status (*renewed*).

To meet the project's analytical and predictive objectives, the base was enriched through **feature engineering**. Several derived variables were constructed, including:

Pricing and Duration Variables

- *nb_jour_couv*
- *prime_par_jour*
- *prime_annualisee*
- *log_prime*
- *anciennete_contrat_jours*
- *anciennete_client_en_jours*

Contractual and Behavioral Variables

- *is_avenant*
- *is_affaire_nouvelle*
- *is_terme*
- *nb_impayes*
- *retard_paiement_moyen_jours*

Risk and Claims-Related Variables

- *nb_sinistres_passe*
- *cout_sinistres_passe*
- *claim_frequency*
- *average_claim_cost*
- *loss_ratio*
- *severity_rate*

Scores and Advanced Indicators

- *client_risk_score*
- *client_profitability*
- *technical_margin*

Following this enrichment phase, the final base includes 44 variables, which constitutes the foundation used for descriptive analysis, the NLQ engine, insight generation, and predictive customer scoring and risk models.

7 General Approach and Architecture

The project is based on 4 mandatory pillars:

1. Data Preparation & Quality
2. Intelligent NLQ Engine (we chose OPENIA)
3. AI Insight
4. Predictive Models & Customer Scoring

and other complementary ones for intelligent data processing, etc.

7.1 Data Preparation & Quality

The data processing module represents the analytical heart of the platform dedicated to the insurance sector. Its architecture is designed to transform raw data into structured and actionable information through a rigorous scientific methodology. The system systematically verifies data presence before granting access to functionalities, thus ensuring operational integrity. It prioritizes the use of already processed data, while retaining the ability to work on original data, ensuring optimal workflow flexibility.

Initialization relies on a scientific processing engine, the `DataProcessingEngine` class, whose state is persisted throughout the user session. This approach maintains analytical context consistency and optimizes performance by avoiding unnecessary reinitializations. All functionalities are organized into four distinct phases accessible via a tabbed interface, offering a logical progression of processing.

The first phase, scientific analysis, deploys a battery of advanced statistical tests to automatically characterize data nature. It identifies variable types, evaluates distributions via normality tests like Shapiro-Wilk and Anderson-Darling, and generates complete metrology including quality and completeness indicators. Each variable benefits from a detailed statistical profile presenting its fundamental characteristics.

The second phase concerns intelligent preprocessing, offering three adaptive strategies: a conservative approach prioritizing original data integrity, a balanced strategy seeking an optimal compromise, and an aggressive method oriented towards machine learning preparation. This processing includes missing value handling, anomaly detection, and adaptive normalization, all with a data type preservation system ensuring semantic consistency.

The third phase focuses on automatic target variable detection, combining semantic analysis and statistical criteria. The engine examines column terminology, evaluates cardinality and distribution balance, and filters variables poorly suited for modeling. Personalized recommendations and adapted visualizations are generated to guide selection.

The fourth phase offers complete statistical exploration with a high degree of customization. Users can filter, sort, and explore data via detailed statistics and interactive visualizations adapted to each variable type. Export functionalities allow saving results in standardized formats for documentation or external integration.

Technically, the module features sophisticated state management preserving intermediate results between processing steps. Its error management system provides clear user feedback while retaining technical information for debugging. The interface is optimized for intuitive navigation with immediate visual feedback and progress indicators.

The whole orchestrates a sequential yet flexible processing pipeline, transforming data processing into a guided scientific process. This methodical approach accompanies the user from initial exploration to preparation for advanced modeling, while ensuring traceability and reproducibility of applied treatments, thus meeting the specific requirements of the insurance domain.

7.2 NLQ Engine

Revolutionizing Data Access for Competitive Advantage

The NLQ (Natural Language Query) Engine is much more than a simple search engine; it is a strategic catalyst. By transforming complex data access into an intuitive conversational interaction, it allows every user, regardless of technical level, to extract crucial information. This ability to query your data and metadata in natural language translates into operational efficiency gains, accelerated decision-making, and undeniable competitive advantage, transforming queries into actionable insights.

- **Expanded Data Access**: Providing tools allowing teams to query data without advanced technical skills, to reduce dependencies on specialized teams and accelerate information exploration.
- **Database Exploitation**: Using existing data to produce statistical analyses, indicators, and predictive models, to exploit available information more completely.
- **Data-Driven Decision Support**: Producing structured and contextualized responses from available data, to support decision processes with measurable and verifiable elements.

Catalyst for Performance and Competitive Advantage

The NLQ Engine is strategically designed to transform how your organization interacts with its data, offering key functionalities that translate directly into tangible return on investment (ROI). It does not just answer questions; it generates contextual explanations and optimized response formats, essential for deep understanding and informed decision-making, thus propelling operational efficiency and competitiveness.

Optimizing Insights and Responsiveness

- **Impactful Executive Summaries**: Short and concise responses for rapid identification of critical trends and strategic decision-making without delay.
- **In-Depth Detailed Analyses**: Exhaustive and nuanced reports, enabling precise diagnostics and strategy development based on verified data.
- **Interactive Demonstrations**: Visualize and manipulate data in real-time for better understanding of correlations and identification of hidden opportunities, accelerating solution adoption.

This unique ability to directly integrate your data and metadata ensures maximum operational flexibility and unprecedented query personalization, allowing you to precisely tailor analysis to your business objectives and discover insights that guarantee lasting competitive advantage.

Security and Confidentiality: A Strategic Competitive Advantage with the NLQ Engine

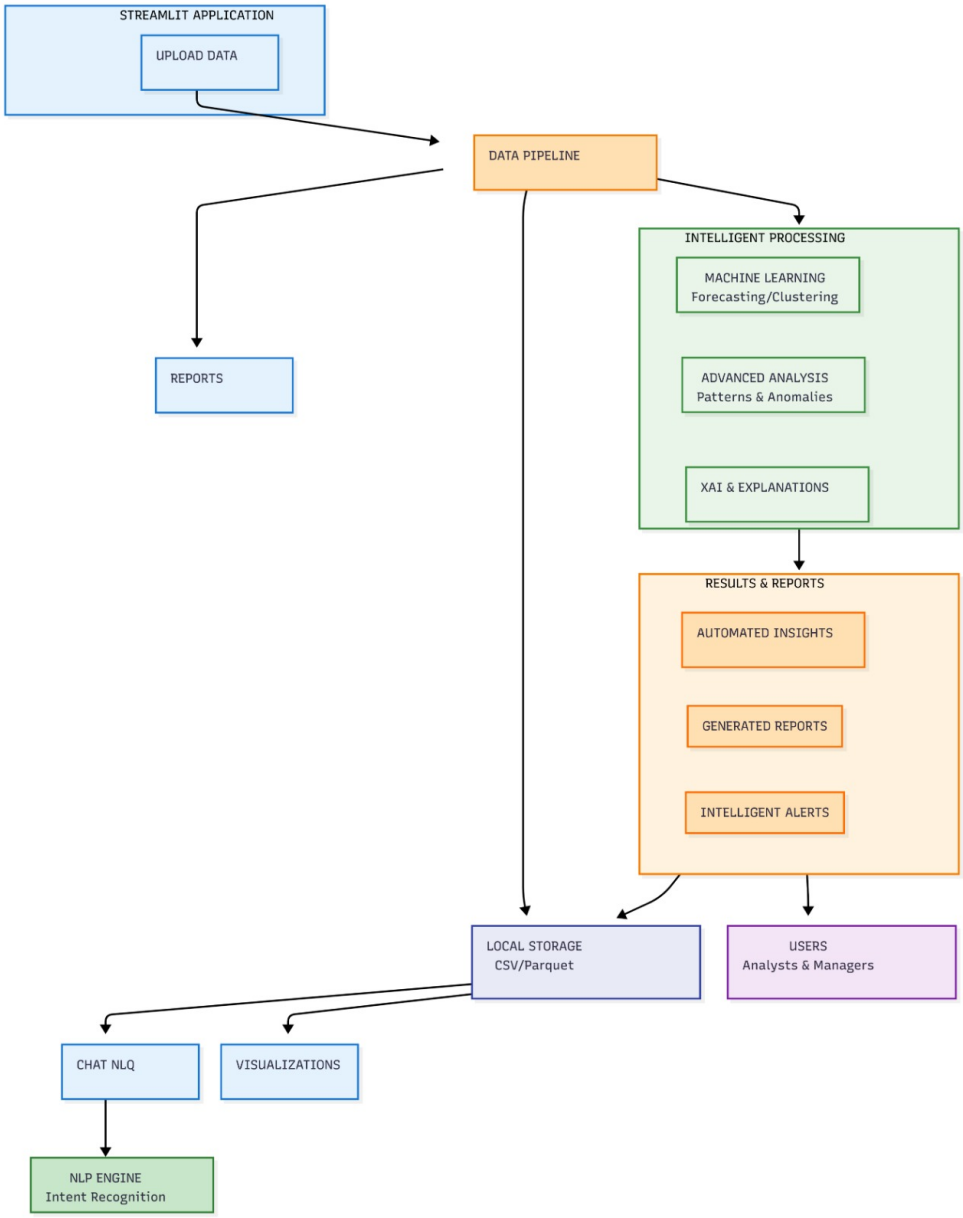

Figure 1: Application Architecture

Data security and confidentiality are no longer mere requirements, but fundamental pillars for company growth and reputation. At the heart of the NLQ Engine design lies a firm commitment to protect your information assets at every stage of their exploitation.

It is within this logic that the NLQ Engine is based on a fundamental principle: strict separation between sensitive data and exploitable data. When using the application, two types of inputs are systematically defined:

1. The original, raw, and sensitive data is initially provided to the system to allow it to understand the context, structure, and associated business logic. This data is used solely as a reference for understanding. Once this phase is completed, it is immediately isolated, then deleted from active processing spaces, to guarantee confidentiality and limit any unnecessary exposure. The system does not retain the

raw data itself, but only its logical and structural history: relationships, schemas, dependencies, implicit rules, and associated metadata. It is on this basis that the engine builds its reasoning. In other words, the chat never reasons from the sensitive data itself, but from the informational footprint it left: its structure, organization, relationships, usage patterns.

This approach ensures that:

- confidential information is never directly exposed.
- business logic remains exploitable.
- reasoning continuity is ensured.
- conversational history remains coherent and relevant.
- confidentiality is preserved long-term.

Thus, even after deleting sensitive data, the system continues to function effectively based solely on the secured logical history, guaranteeing both performance, traceability, and protection of critical information.

2. A metadata layer, on which all processing, analyses, and interactions will be performed.

This architecture ensures that operations are never performed directly on critical data, but only on their secured representation. This mechanism drastically reduces risks of leakage, exposure, or mishandling, while maintaining complete and performant analysis capability.

We deploy advanced security protocols and robust architectures to guarantee maximum protection of all sensitive information. This approach ensures not only rigorous compliance and significant risk reduction, but also strengthens trust from your clients and partners, transforming security into a true value lever and a lasting competitive advantage.

- **Protection of Critical Assets**: Implementation of advanced security mechanisms based on isolation of sensitive data and their indirect exploitation via secured metadata. This approach enables proactive protection of strategic information and intellectual property, while ensuring their integrity and availability for operational continuity.
- **Optimized Regulatory Compliance**: Native integration of regulatory requirements (GDPR, HIPAA, etc.) through an architecture that limits direct exposure of personal and sensitive data. This structural separation between source data and exploited data reduces legal and financial risks, while facilitating access to new markets.
- **Secure Access Management**: Reinforced authentication and fine-grained privilege management for each interaction. Access to sensitive data is strictly controlled, while users work exclusively on secured representations (metadata). This guarantees confidentiality, prevents any non-compliant use, and protects business process integrity.

7.3 Insight AI

The Insight Engine is designed to transform raw contract data into a clear and directly exploitable reading of **customer risk**. Concretely, it first consolidates information at the **customer level** to produce a reliable steering dashboard, then calculates a **normalized risk score out of 100**, automatically classifies each customer as **Low / Medium / High**, and finally generates a **textual explanation** of the factors justifying this risk level. The objective is to move from a descriptive analysis to a decision-oriented logic, where the user can quickly understand *who is at risk, why, and which action to prioritize.*

Operation relies on a first **consolidation** step: from operations and contracts, the engine aggregates information by customer (`ncli`, `nomncli`) and builds synthetic indicators. We obtain notably **total premium**, **average premium** and **last premium**, as well as measures on **coverage duration** (average, minimum, maximum and variability). The engine also captures contractual behavior through the number of operations and the number of **endorsements** (detected via the `libop` field), and integrates, if the columns exist, essential signals like **payment delays**, **unpaid amounts**, **claims history** (frequency and costs), and even **technical profitability** indicators such as margin and **loss ratio**. Particular attention is paid to robustness: missing values, divisions by zero, and infinite values are handled automatically, then a filling strategy (medians or zero depending on variables) ensures an exploitable table even when data is imperfect. A central indicator is built at this stage: **premium per day**, calculated as `Prime / nb_jour_couv`, then averaged at customer level (`prime_par_jour_moy`).

Once the customer table is built, the engine calculates a **risk score**. Variables are first **normalized** to the interval ([0,1]) (min-max), then combined into a final score scaled to **0 to 100**. The score mixes structural factors (always present) and optional factors (used only if they exist in the base). The score base relies on **average premium per day**, **contractual instability** via the number of endorsements, and **coverage duration**, transformed into a risk factor so that shorter durations increase risk. Then, when data allows, the score incorporates additional dimensions particularly relevant for insurance: **claims frequency and cost**, **payment delays**, **unpaid amounts**, **historical termination rate**, and **loss ratio**. The result is then segmented into three categories: **Low (0–30)**, **Medium (30–70)**, and **High (70–100)**. To each level correspond an **action priority** (Low/Medium/High) and a **business recommendation** automatically formulated (e.g., maintenance and retention for stable profiles, enhanced monitoring for medium risk, or corrective actions and pricing review for high risk).

To make the score understandable and credible, the Insight Engine adds an **explicability** layer: for each customer, it generates an insight sentence based on simple, interpretable rules. The explanation mentions detected signals, such as a **high premium per day** (compared to portfolio median), a **short average duration**, **contractual instability** (several endorsements), **high claims frequency**, **frequent payment delays**, or an **unfavorable loss ratio**. If no notable signal appears, the engine concludes that the customer presents a stable profile. This approach transforms a "technical" score into a message directly understandable by a non-technical decision-maker.

Beyond the score and individual insight, the engine also produces a portfolio reading: it calculates aggregate indicators (number of analyzed customers, average score, distribution

by risk level, shares of multi-claimants, significant delays, etc.) and makes interactive visualizations available to explore results. The user can examine an isolated variable (distribution, boxplot, statistics), analyze relationships between variables (scatter plot, correlation, comparison by category), and go further with multivariate analyses: **PCA** to understand variable structure and explained variance, **MCA** to explore relationships between categorical modalities, or **K-means clustering** to automatically segment the portfolio and characterize homogeneous customer groups. Finally, the engine is capable of generating a **narrative report** intended for decision-makers, structured around a portfolio summary, key risk factors, strategic recommendations, and follow-up KPIs.

The main interest of this component is that it combines **automation**, **interpretability**, and **actionability**: it consolidates data, produces robust scoring, explains results in clear language, and guides business decisions. Technically, the implementation relies on `pandas` and `numpy` for aggregation and indicator engineering, `plotly` for interactive visualizations, `scikit-learn` for normalization, PCA and clustering, `scipy.stats` for some statistical tests, `prince` for MCA (if installed), and potentially `statsmodels`/`matplotlib` for analyses like ACF when time series are used.

7.4 Predictive Models

Our suite integrates cutting-edge predictive models, meticulously developed to decipher emerging trends and anticipate future behaviors. By exploiting the entirety of your historical data, these models deliver projections of unparalleled accuracy, giving you a decisive strategic advantage for proactive decision-making, resource optimization, and sustainable growth. It is the essential tool to transform uncertainty into opportunity, ensure tangible return on investment, and consolidate your market leadership position.

> Anticipating the future is no longer conjecture, but a strategic mastery of data to sculpt your success and outpace the competition.

Unlock Future Growth: The Mechanics of Our Predictive Models

Our cutting-edge predictive models transform your historical data into a strategic roadmap. Relying on sophisticated algorithms, they detect latent patterns, correlations, and causalities, offering accurate projections to anticipate sales, optimize customer strategies, and mitigate operational risks. This ability to visualize the future gives you a decisive competitive advantage, ensuring informed decisions and maximum return on investment.

The modeling process is organized into three distinct and complementary phases, each addressing a specific dimension of predictive analysis.

Exploratory analysis serves as the foundation for the entire modeling process. This phase is not limited to simple visualization; it implements automatic characterization of potential variables. The system examines each column to determine its statistical nature: it distinguishes binary categorical variables (2 classes), multi-class variables (up to 20 classes), and continuous numerical variables. For each detected type, it generates adapted

visualizations: pie charts for binary variables, bar charts for multi-class, and histograms with descriptive statistics for numerical variables. Correlation analysis goes beyond traditional representations by providing an interactive heatmap accompanied by a hierarchical ranking of the ten strongest associations detected in the data. This approach allows the user to quickly identify key relationships and select relevant predictive variables for later stages.

The classification module implements a sophisticated processing pipeline inspired by machine learning best practices while maintaining remarkable accessibility. Data preparation begins with rigorous input validation, automatically excluding unusable variables like dates or columns with too many categories. The system applies intelligent one-hot encoding to categorical variables, with an automatic limit on the number of categories to avoid dimensional explosion. Missing value management offers three configurable strategies: median imputation, mean imputation, or removal of incomplete rows. A variance filtering step automatically eliminates quasi-constant variables that would provide no discriminative information.

Model training for classification supports four main algorithms, each optimized for the insurance domain. The Random Forest Classifier is implemented with hyperparameters adapted to medium-sized data, including automatic class balancing mechanisms via the `class_weight` parameter. XGBoost offers advanced performance with native handling of imbalanced data via `scale_pos_weight`. Logistic regression provides an interpretable option with L2 regularization, while Gradient Boosting offers a balance between performance and complexity. Each model benefits from automatic hyperparameter optimization via adaptive grid search that adjusts search complexity according to available data size.

Model evaluation for classification goes beyond basic metrics. Beyond accuracy, precision, recall, and F1-score calculated with weighting adapted to the problem type, the system generates interactive confusion matrices with thermal visualization. For binary classification problems, it produces ROC curves with AUC calculation and precision-recall curves. Variable importance, when available (notably for tree-based models), is presented as ranked horizontal bar charts and a detailed exportable table. The module also includes error analysis with sampling of misclassified cases for deeper investigation.

The regression module addresses the prediction of continuous variables with a methodology adapted to the specifics of insurance data. Data preparation includes specific outlier treatment with three configurable approaches: preservation to maintain data integrity, removal to eliminate potentially harmful extreme values, or winsorization to limit their influence while retaining information. Normalization offers four methods: standardization (suitable for Gaussian data), min-max (for bounded data), robust (resistant to outliers), or no transformation for algorithms insensitive to scale.

Implemented regression algorithms include Random Forest Regressor with parameterization adapted to numerical prediction problems, XGBoost Regressor optimized for computational performance, linear regression for its interpretability and speed, and Gradient Boosting Regressor as a balanced alternative. Each algorithm benefits from specific hyperparameter optimization: search for optimal depth and number of trees for ensemble methods, optimization of learning rate and regularization for boosting methods.

Model evaluation for regression uses a comprehensive battery of metrics adapted to numerical prediction problems. R^2 measures the proportion of explained variance, RMSE and MAE quantify absolute error with different outlier sensitivities, and MAPE expresses relative error for inter-dataset comparisons. Visualizations include scatter plots comparing predictions and actual values with an ideal reference line, error distribution histograms, and QQ-plots to check residual normality. Variable importance is calculated for tree-based models and presented hierarchically.

The underlying technical architecture ensures consistency and reproducibility of analyses. The system maintains a complete state of each modeling in the user session, allowing navigation between different phases without loss of information. Data preparation functions include robust validations and fallback mechanisms to handle edge cases. Model export saves not only the trained algorithm but also all transformations applied to the data, ensuring the same preprocessing will be applied during production use. Automatic code generation for model reproduction facilitates transfer to deployment environments and ensures complete traceability of the analytical process.

This modular and methodical approach transforms predictive modeling from a complex technical task into a guided process accessible to business analysts while retaining the rigor needed by data scientists. It integrates recent theoretical advances in machine learning while remaining grounded in the practical needs of the insurance sector, thus offering a complete platform from data exploration to deployment of robust and interpretable models.

Maximize Performance and Profitability with Predictive AI

The strategic integration of predictive models is no longer an advantage, it is an imperative necessity for any company aiming for operational excellence and sustained growth. Our AI suite transforms your raw data into actionable information, allowing you to radically optimize resource allocation, refine marketing campaigns for unprecedented results, and anticipate risks for better mitigation. By offering a deep understanding of future trends, our solutions guarantee maximum return on investment (ROI), reduce operational costs, and propel customer experience to unparalleled heights. It is the key to transforming your data into a lasting competitive advantage, driving exponential growth, and establishing your market leadership.

We integrated several models into the application, allowing the user to have a wide choice to better adapt their study.

Random Forest

Random Forest is an ensemble learning model that combines a large number of decision trees built independently and randomly to improve prediction quality. Its operating principle relies on intentionally introducing randomness at two levels: on one hand, each tree is trained on a bootstrap sample of the data, and on the other hand, at each node of the tree, only a random selection of variables is considered to determine the best split, generally by maximizing an impurity reduction (like quadratic error in regression). Once trees are built, the final prediction is obtained by aggregating individual predictions, as

an average in regression or majority vote in classification. Mathematically, the forest approximates the target function by averaging tree predictions, which strongly reduces variance while keeping bias low. This structure makes Random Forest particularly robust to nonlinearities, complex variable interactions, and data noise, explaining its excellent practical performance, for example for predicting customer risk in insurance, where each tree learns different rules and the forest provides a final stable and reliable score.

XGBoost

XGBoost (eXtreme Gradient Boosting) is a **regularized gradient boosting tree method**, initially proposed by Chen and Guestrin (2016), which has established itself as one of the most performant algorithms for predictive analysis on large datasets. Its principle relies on the sequential construction of regression trees, each new tree being added to correct errors of previous trees, while minimizing a **regularized objective function** combining a convex loss function and complexity penalties on tree structure. Optimization is performed using a **second-order Taylor approximation**, exploiting both gradients and hessians of the loss function, enabling efficient and precise optimization.

However, the article by Yang Guang [3] highlights an **important theoretical limitation** of classic XGBoost: the requirement of convexity of the loss function, a necessary condition to guarantee algorithm convergence. Yet, in many real applications, notably in **non-life insurance**, variables to model follow asymmetric or heavy-tailed distributions, for which non-convex loss functions (from parametric likelihoods) are more appropriate. To address this limitation, the author proposes a **generalization of XGBoost**, which relaxes the convexity constraint and allows the use of more general loss functions, provided they are twice differentiable and possess a unique minimum.

Finally, the article extends XGBoost to a **multi-parametric framework**, in which several parameters of the same distribution (for example mean and dispersion) are estimated simultaneously via distinct but coordinated trees. This extension brings XGBoost closer to distributional statistical models while retaining the flexibility of machine learning methods, thus offering a powerful framework for probabilistic modeling and insurance pricing.

Logistic Regression

Logistic regression is a probabilistic model intended to explain a binary variable $Y \in \{0, 1\}$ from explanatory variables X. It models the conditional probability:

$$P(Y = 1 \mid X) = \frac{1}{(1 + e^{X\beta})}$$

This formulation relies on the hypothesis that the log-odds (logarithm of the probability ratio) is a linear function of the explanatory variables:

$$\log\left(\frac{P(Y=1 \mid X)}{1-P(Y=1 \mid X)}\right) = X\beta$$

Gradient Boosting

Gradient Boosting is an ensemble learning method consisting of building a predictive model in the form of an additive sum of weak models, generally decision trees, adjusted sequentially. Unlike ensemble methods based on independent aggregation (like random forests), Gradient Boosting adopts an iterative and constructive logic: at each step, a new model is trained to correct errors made by the ensemble of previous models. This approach finds its theoretical foundation in numerical optimization, the learning problem being formulated as the minimization of a loss function in function space.

The article shows that Gradient Boosting can be interpreted as a functional gradient descent, where each new model is adjusted to be strongly correlated with the negative gradient of the loss function evaluated on the data. When the loss is quadratic, the procedure reduces to successively adjusting residuals, but the general framework allows the use of arbitrary loss functions adapted to regression, classification, or even specific distributions (Bernoulli, Poisson, survival). This flexibility explains the broad success of Gradient Boosting in many empirical applications.

The authors also highlight that the high predictive power of Gradient Boosting Machines comes with a significant risk of overfitting, due to their ability to approximate complex nonlinear relationships. To control this complexity, several regularization mechanisms are essential: reduction of the learning step (shrinkage), data subsampling (stochastic gradient boosting), and early stopping. Well-parameterized, Gradient Boosting achieves an optimal bias-variance compromise, making it one of the most performant algorithms for modern predictive analysis.

8 Project Timeline

Our agile work plan, structured over a six-day sprint (from January 13 to 19, 2026), is designed to guarantee the incremental delivery of a functional solution with high business value. The project begins with an essential day of scoping and foundations, during which Ibrahima, as team leader and responsible for the NLQ engine, defines with the team the vision, technical architecture, and common communication interfaces, ensuring all modules can communicate. Mariam, responsible for the insights engine, identifies key business indicators and prepares data, while Babacar, in charge of the predictive model, selects the initial scoring algorithm. For her part, Aya designs the first user interface mockups. This initial phase is crucial to align the team on indispensable "Must" functionalities.

The second and third days are dedicated to parallel development and integration of basic components. Each member works simultaneously on their core expertise: Ibrahima develops the natural language understanding engine and the API Gateway; Mariam builds the

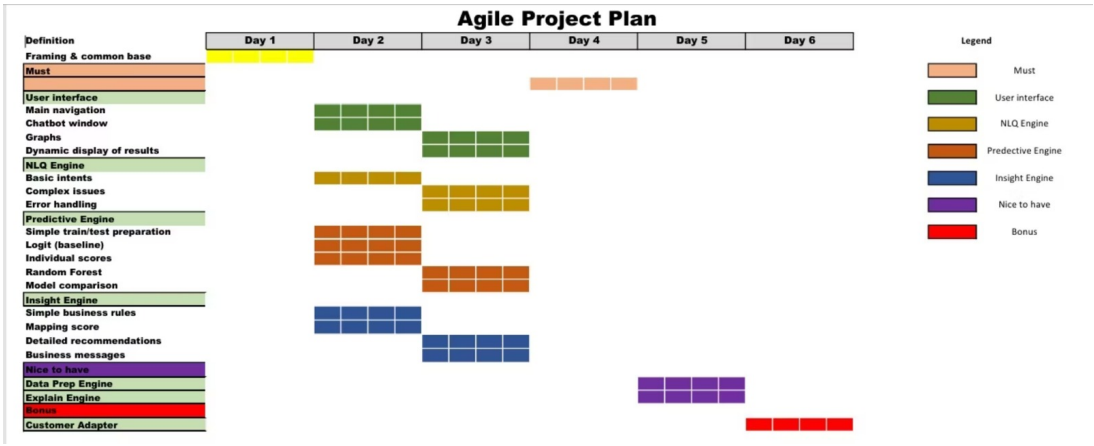

Figure 2: Agile Project Plan

data cleaning pipeline and insight generator; Babacar trains and deploys his prediction model; and Aya constructs the chat and visualization interface. The objective of the third day, "Must consolidation," is to integrate all these blocks to obtain a perfectly functional and stable minimum viable product (MVP). This parallel and integrative approach allows us to test each component progressively and avoid bottlenecks.

Once the core is stable, the fourth and fifth days are dedicated to "Nice to Have" improvements and "Bonus" functionalities. Ibrahima can enrich the NLQ with conversational context, Mariam adds an alert system and advanced visualizations, Babacar integrates model explainability, and Aya fine-tunes the user experience with animations and a presentation mode. These additions, conditioned on core solidity, bring real added value and a "wow factor" for the demonstration. Finally, the sixth day is entirely dedicated to preparing and rehearsing a compelling presentation, where each presents their contribution in a coherent narrative highlighting the business value of our solution.

9 Choice of Tools, Technologies, and Packages

For this project, we chose a modern, lightweight technology stack focused on data & AI, enabling rapid, modular, and easily demonstrable development, while remaining close to professional standards used in enterprise with varied and especially secure functions.

Main Language: Python

We chose `Python` as the main language for the following reasons:

- science and artificial intelligence;
- richness of the library ecosystem;
- rapid prototyping, essential in a hackathon;
- ease of integration between data, models, and interfaces.

- strong adoption in the data field

As a coding environment, we chose *Pycharm Community Edition* 2025 with the version and directly on a member's Github `3.14, 3.13, and 3.12 of python`.

```
1  import streamlit as st
2  import pandas as pd
3  import numpy as np
4  import plotly.express as px
5  import sys
6  import os
7  import warnings
```

Figure 3: Some packages

The module (`numpy`,`pandas`) allows explaining scores produced by models, identifying the most influential variables, and transforming technical results into clear business recommendations. The objective is to strengthen decision-makers' trust in results produced by artificial intelligence.

`Streamlit` was chosen for its simplicity and ability to quickly produce interactive interfaces. Visualization libraries allow representing key indicators, scores, and model results in understandable graphical form, thus facilitating business interpretation of analyses (`plotly`,`seaborn`).

Several models were tested, notably Random Forest and Gradient Boosting:`xgboost`, `scikit-learn`, `joblib`, etc. These algorithms are well adapted to classification problems, like predicting contract renewal or termination, and offer a good compromise between performance and interpretability.

For training the search engine "CHATBOT", we chose to use OpenAI to generate the API key that works with *CHATGPT4 Turbo*.

Thus, everything done remains local and this allows managing the "security" aspect of the application.

10 Development Challenges and Solutions

Difficulties encountered during the development of the intelligent analytics application

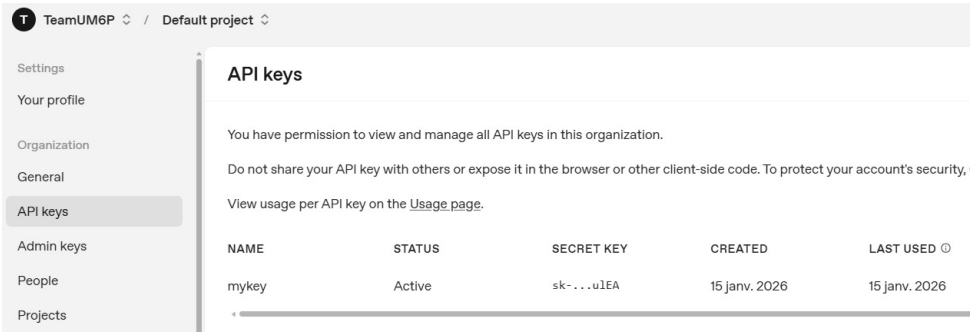

Figure 4: Generation: API key

10.1 Challenges of Conversational AI

One of the first major difficulties we encountered concerns the development of the intelligent chat system. Initially, the natural language processing models we implemented provided irrelevant responses, often far removed from the business context of our application. Response times were particularly long, regularly exceeding ten seconds, which made the user experience frustrating and unnatural. We had to face the reality of the technical limitations of available open-source solutions, which did not meet the accuracy and speed requirements necessary for a professional application.

This situation forced us to opt for a paid API, which offered significantly superior performance. However, this solution introduced its own challenges: the initial API key expired in the middle of development, and its renewal created unexpected service interruptions.

10.2 Architectural Complexity and Component Integration

Designing the overall architecture presented considerable challenges, particularly in articulating the different modules. The Insight Engine, the analytical heart of our application, required particularly complex development. Transforming raw statistical results into actionable and understandable insights demanded the creation of sophisticated pattern detection algorithms and automatic explanation generation. Implementing the embedding system, essential for representing business concepts in a consistent vector space, proved more technical than anticipated.

Assembling the different files and modules constituted an additional difficulty. Problems with circular dependencies and library incompatibilities appeared regularly. The modular structure we initially envisioned had to be redesigned several times to allow independent evolution of components while maintaining overall coherence.

10.3 Selection and Optimization of Machine Learning Models

The modeling phase confronted us with complex algorithmic choices. We undertook an extensive literature review to identify the methods most suited to our specific problems.

Random Forest was selected for its robustness with heterogeneous data, while XGBoost showed remarkable performance on complex prediction problems. Logistic regression, despite its apparent simplicity, remained in our arsenal for its interpretability and execution speed, qualities essential for certain real-time applications.

Training these models revealed significant computational constraints. Training times on our voluminous datasets became prohibitive, requiring advanced optimizations and sometimes compromises on model depth. These challenges led us to implement techniques such as feature selection, dimensionality reduction, and distributed computing approaches to make the training process more efficient while maintaining model accuracy.

References

[1] NARSA. *Annual Report 2022 on Road Safety in Morocco*. National Road Safety Observatory.
https://www.narsa.ma/sites/default/files/2024-11/Rapport%20de%20la%20SR%202022%20V5_231020_140005_compressed.pdf

[2] Misha Denil & al, *Narrowing the Gap: Random Forests In Theory and In Practice*.
https://proceedings.mlr.press/v32/denil14.pdf

[3] Yang Guang, *Generalized XGBoost Method*, 2022.
https://arxiv.org/abs/2109.07473

[4] Hosmer, Lemeshow & Sturdivant (2013), *Applied Logistic Regression*, Wiley.
https://books.google.co.ma/books?hl=fr&lr=&id=bRoxQBIZRd4C&oi=fnd&pg=PR13&dq=Hosmer,+Lemeshow+%26+Sturdivant(2013),+Applied+Logistic+Regression,+Wiley

[5] Natekin, A., & Knoll, A. (2013). *Gradient Boosting Machines, a Tutorial*. Frontiers in Neurorobotics, 7:21.
https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2013.00021/full