Harnessing AI for Business: Data Scientists

Abdoulay Lashley, Bhavya More, Dante Caraballo,
Goutham Korada, Reetika Madan, Vladimir Martirosyan

BUDT751

May 17, 2025

Introduction

Data scientists play an essential role in aiding organizations in extracting actionable insights from complex data. Applying machine learning, they build predictive models that inform business strategy, optimize operations, and enhance the customer experience. This project aims to automate some of the machine learning process and enhance it using AI, specifically model selection and running, as well as SHAP explainability, to extract insights from structured data.

Today, the ability to combine statistics with scalable ML tools is essential. This project reflects how data science and AI integrate into business decision-making. The methods presented mirror real-world applications used by data science professionals in finance, healthcare, retail, and logistics.

General Overview of Role / Business

A data scientist operates at the intersection of data engineering, statistics, and business analytics. Typically, they are responsible for cleaning and structuring raw data, applying supervised or unsupervised learning algorithms, and showcasing findings through visualization and metrics. In sectors like transportation, banking, or e-commerce, data scientists are often assigned tasks that involve the optimization of customer segmentation, fraud detection, demand forecasting, and risk modeling. Our project reflects a data science workflow using clean data, domain understanding, and ML to generate meaningful insights and conclusions. The business value lies in enabling AI-enhanced, transparent, scalable decision systems based on historical data and statistical analysis.

Methods and Methodologies

Our project closely mirrors the job functions of modern data scientists who frequently work with end-to-end pipelines, from preprocessing and model training to explainability and bias evaluation. The role demands a blend of statistical rigor, machine learning implementation, and business acumen. In this context, we utilized Shapley Additive exPlanations (SHAP) to interpret complex model predictions, aligning with how AI is increasingly expected to deliver transparent and justifiable decisions. Moreover, we employed LLaMA (LLaMA-3-8B Instruct) for natural language explanations, showcasing how AI can support communication between data scientists and non-technical stakeholders. As businesses move toward responsible AI, our methodology also explores prompt engineering and model fine-tuning strategies to mitigate hallucinations, a growing concern in generative AI. Looking ahead, integrating model monitoring tools, automated retraining pipelines, and human-in-the-loop evaluation could further future-proof AI deployments.

Product Overview

Our product is a user-friendly, Explainable AI web application designed to democratize model interpretation using SHAP. Key features include the ability to upload a CSV dataset, select a target column for analysis, and receive clear visual and textual explanations for model decisions. The platform incorporates a dynamic user input system that prompts the user to identify whether the target variable is categorical or numerical. Based on this selection, the tool recommends appropriate models: Decision Tree or XGBoost Classifier for classification problems, and Linear Regression or XGBoost Regressor for regression tasks.

The system leverages SHAP values to highlight the most influential features behind each prediction, making complex models more interpretable. Users can optionally specify a row to generate individual-level insights, enhancing transparency at a granular level. Additionally, the interface is designed to support future integrations such as customizable model selection, multi-language support, and improved natural language feedback using large language models (LLMs). This tool is especially valuable for professionals in healthcare, finance, and operations who require both interpretability and performance from machine learning models.

Technical Implementation

Our tool is designed to work with any cleaned data set (null values handled and necessary features imputed or encoded). Raw data that can be input to the tool once cleaned can be found on GitHub. For demonstration purposes, we used the Titanic train data. It included columns such as:

PassengerId: *INT to identify customers (index)*

Survived: *Survived or not: 0 = No, 1 = Yes*

PClass: *Ticket class: 1 = 1st, 2 = 2nd, 3 = 3rd*

Name: *Name of the Passenger*

Sex: *Gender*

Age: *Age in Years*

SibSp: *Number of siblings / spouses aboard the Titanic*

Parch: *Number of parents / children aboard the Titanic*

Ticket: *Ticket number*

Fare: *Passenger fare*

Cabin: *Cabin number*

Embarked: *The port the ship embarked on. Port of Embarkation: C = Cherbourg, Q = Queenstown, S = Southampton*

We observed missing values in the “age,” “cabin,” and “embarked” columns. We decided to impute “age” with the median and “embarked” with the mode. Additionally, “cabin,” “name,” and “ticket” were dropped from the data frame. We then encoded categorical variables like “sex” using the Label Encoder.

In addition to building the model manually, we implemented an interactive user input system that allows users to upload a dataset and specify a target variable. The user selects whether the target is categorical or numerical and dynamically offers the appropriate modeling options. For categorical targets, users can choose between a Decision Tree and an XGBoost Classifier. For numerical targets, options include linear regression and XGBoost Regressor. This logic-driven model selection enhances usability and expands the product's flexibility across a broader range of datasets.

Model Performance and Optimization

To evaluate the effectiveness of our machine learning model, we began by establishing a baseline using the default configuration of a Black Box (XGBClassifier). Initially, we observed a training accuracy of approximately 95% and a test accuracy of 79%, indicating strong initial performance but also suggesting possible overfitting.

Next time, we would like to optimize several hyperparameters, including max_depth, n_estimators, reg_alpha, and reg_lambda, to improve generalization while maintaining model complexity. We would use a validation split to monitor model performance and avoid overfitting during training. Due to computational constraints in the Google Colab environment, further tuning would be conducted through manual adjustments rather than full cross-validation. However, SHAP values were used to evaluate feature importance, which in turn helped us ensure the model was not relying heavily on biased or uninformative variables.

To explore further model improvement, we also considered logistic regression as a comparison benchmark. Although simpler, logistic regression produced lower accuracy and lacked the decision-tree interpretability offered by SHAP when paired with XGBoost. These comparative insights validated our model selection and tuning strategy. Incorporating automated hyperparameter tuning methods like grid search or Bayesian optimization would be a natural extension for improving performance in a scalable and reproducible manner.

Bias Detection and Evaluation

The dataset can create data bias, which is reasonably simple to mitigate. Before the model can be ingested, we must ensure that all data is clean (e.g., Null Values Handled, Necessary Columns Encoded).

Currently, the pretrained Llama model is hallucinating when given our data. Many factors could cause this. However, we can mitigate this by refining the training dataset or, in our case, using a Llama model with more parameters. Additionally, we could further prompt engineer to refine the model's output and apply a form of external validation or self-evaluation. Furthermore, practices like hyperparameter tuning would help the model become more accurate and less hallucination-prone.

Future Implementation

Looking ahead, we envision expanding the capabilities of our Explainable AI platform in several impactful ways. First, we aim to integrate multiple model types beyond XGBoost, such as logistic regression, random forests, and deep learning models, allowing users to compare results across algorithms. Additionally, future versions will include dynamic SHAP visualizations, real-time model updates based on new data, and user-specific model retraining options. To improve accessibility, we plan to add support for multilingual natural language explanations, voice-to-text input, and integrations with enterprise data platforms such as Snowflake or AWS S3. To further enhance interpretability, users will eventually be able to toggle between global model explanations and local prediction breakdowns. Finally, embedding audit logs and compliance-friendly documentation features will make this tool suitable for industries with strict regulatory demands.

In 5–10 years, we envision this solution becoming a plug-and-play tool embedded in enterprise software, allowing analysts and managers to interpret ML models without deep technical expertise. As regulatory pressure for explainable AI grows, platforms like this could serve as audit-compliant tools for transparent decision-making in finance, healthcare, and HR.

Challenges

During development, we encountered several key challenges that shaped the scope and direction of our work. One significant issue was model hallucination when using LLaMA to generate SHAP explanations. The LLM occasionally produced plausible-sounding but incorrect outputs, highlighting the importance of prompt design and model size selection. Additionally, preprocessing the Titanic dataset required careful handling of missing values and inconsistent formats, issues that are often amplified in real-world datasets. We also struggled with balancing SHAP visual clarity and computational efficiency, particularly when scaling explanations to multiple rows. Technical constraints within Google Colab (e.g., limited GPU availability and memory) occasionally slowed down model training and inference. Lastly, ensuring that our platform remained user-friendly while incorporating advanced ML features presented an ongoing UI/UX design challenge.

Conclusion

This project demonstrates the power of combining machine learning with explainable AI to create transparent, user-centric tools for data-driven decision-making. By integrating an XGBoost classifier with SHAP value visualizations and LLaMA-generated natural language explanations, we developed a platform that not only predicts outcomes but also communicates the “why” behind each prediction. Our work reflects real-world practices used by data scientists to ensure fairness, accountability, and usability in AI systems. While we encountered challenges related to model hallucination and computational constraints, we addressed them through prompt refinement, data preprocessing, and modular design. The result is a scalable and intuitive product that can be adapted to various industries where trust and interpretability are essential. As AI continues to evolve, our solution provides a foundation for responsible innovation in machine learning applications.

References

1. Hegazy, Y. (2022). *Titanic - Machine Learning from Disaster Dataset*. Kaggle. https://www.kaggle.com/datasets/yasserh/titanic-dataset