

A Review of Automated and AI-Assisted Exploratory Data Analysis Techniques: Gaps and Directions for an AI-Powered EDA Generator

Bhanu Bhardwaj, Shreya Singh, Prabha Panwar, Saransh Sharma, Sarthaj Singh Rana

Department of Computer Science & Engineering (Data Science)

Inderprastha Engineering College, Ghaziabad

© The Author(s), under exclusive license to publication division, IPEC Journal of Science & Technology, 2026

Abstract—Exploratory Data Analysis (EDA) remains one of the most time-consuming stages of the data science workflow, and a growing body of research has attempted to automate it through statistical libraries, machine-learning-based recommenders, and, more recently, large language model (LLM) driven systems. This paper presents a structured review of twelve representative works spanning rule-based autoEDA packages, ML-guided data exploration, and LLM-based insight and visualization generation. For each category we summarize the underlying methodology and the specific limitations reported or observed, and we consolidate these into a comparative table. Across the reviewed literature, five recurring gaps emerge: absence of natural-language interpretation, dependence on heavy or domain-specific infrastructure, poor scalability to wide or large datasets, fragmented tooling across profiling, visualization and modeling, and limited accessibility for non-programmers. We then discuss how these gaps motivate and inform the design of our own project, the AI EDA Generator, an integrated pipeline that combines automated statistical profiling with an LLM reasoning layer to produce a single exportable, plain-language report. The review is intended to situate the proposed system within the current state of the art and to justify its design choices with reference to documented shortcomings of existing tools.

Index Terms—Exploratory Data Analysis, AutoEDA, Large Language Models, Data Profiling, Automated Insight Generation, Data Visualization

I. INTRODUCTION

Exploratory Data Analysis (EDA) is the preliminary step in which analysts examine a dataset's structure, quality and distributions before formal modeling begins. Industry estimates consistently place 60–80% of a typical data project's timeline in profiling and cleaning rather than modeling itself, and this bottleneck has motivated more than a decade of research into automating parts of the EDA workflow. Early automation efforts focused on statistical and rule-based libraries that compute summary statistics and generate standard charts with minimal user input. More recent work has explored machine-learning-based recommenders that suggest which chart or statistic is likely to be “interesting” next, and the most recent wave of research uses large language models to go a step further: interpreting the computed statistics and expressing them as human-readable insights, or even generating visualization code directly from natural-language goals.

Despite this progress, no single reviewed system offers a complete, low-code pipeline that combines profiling, visualization, natural-language interpretation and reproducible reporting for a general (non-expert) user. This paper reviews twelve representative works from this space, identifies the concrete limitation associated with each, and synthesizes the recurring gaps that remain open. These gaps are then used to motivate the architecture of our ongoing project, the AI EDA Generator, which is briefly described in Section V.

The remainder of the paper is organized as follows. Section II outlines the scope and method used to select the reviewed literature. Section III presents the review, grouped into three

categories: statistical/rule-based autoEDA tools, machine-learning-guided exploration, and LLM-driven insight and visualization systems. Section IV consolidates the findings in a comparative table and discusses the common gaps. Section V outlines how the proposed system addresses these gaps, and Section VI concludes the paper.

II. REVIEW SCOPE AND METHOD

The literature considered in this review was drawn from IEEE Xplore, the ACL Anthology, ACM Digital Library and ar Xiv, using search terms including “automated exploratory data analysis”, “autoEDA”, “LLM data insight generation” and “automated data profiling”. Twelve works were selected to represent three broad, chronologically overlapping generations of automation: (i) statistical and rule-based libraries that compute and visualize summary statistics with little or no interpretation layer; (ii) machine-learning and recommender-based systems that guide the user toward statistically interesting views of the data; and (iii) LLM-based systems that either interpret statistical output in natural language or generate visualization/analysis code directly from a data summary. The goal was breadth across this progression rather than exhaustive coverage of any single sub-area, so that the limitations identified are representative of the field as a whole rather than of one tool family.

III. LITERATURE REVIEW

A. Statistical and Rule-Based AutoEDA Tools

Staniak and Biecek [4] conducted a systematic review of fifteen R packages for automated EDA, cataloguing which stages of profiling, distribution analysis and correlation

detection each package automates. Their survey concludes that while individual packages automate specific sub-tasks well, none of the surveyed tools attempts to automatically summarize or prioritize findings for the user — the analyst must still manually decide which output is significant, and packages are largely non-interoperable with one another.

Putatunda et al. [7] introduced SmartEDA, an R package that produces descriptive statistics and visual summaries for a dataset with a small number of function calls. While SmartEDA reduces the coding burden relative to writing custom scripts, its output remains raw tables and plots; the package does not generate any natural-language explanation of what a given distribution or correlation implies, and it is limited to the R ecosystem.

A conference study on Autoviz [1] evaluated an automated visualization library for classification-oriented machine learning datasets, showing that it can highlight feature relevance to a target variable without manual chart scripting. However, the tool is scoped narrowly to visualization for classification preprocessing; it does not perform missing-value or duplicate-record profiling, and produces no textual interpretation of the charts it generates.

A separate IEEE study [3] proposed a framework for automating anomaly and outlier detection within EDA using Hadoop and Spark for distributed computation. This approach automates a specific and often-skipped part of EDA — outlier detection — at scale, but it requires a big-data cluster deployment, which makes it impractical for a single analyst or student working with a typical CSV-sized dataset.

Finally, a conference paper on the role of EDA in data science [5] surveys the tools and packages commonly used for EDA and documents the practical impact of skipping this phase (for example, undetected data leakage or misleading model performance). This work is valuable as a motivational and cataloging reference but does not itself propose or evaluate an automation architecture.

B. Machine-Learning-Guided and Recommendation-Based Exploration

A broad overview published at SIGMOD [6] surveys the progression of ML-based EDA automation, from recommender systems that suggest a single next chart or query, through active-learning classifiers of user interest, to reinforcement-learning-based fully automated exploration agents. The overview notes that interestingness-prediction models generally need to be trained or calibrated per domain or per user, and that they can be brittle when applied to a dataset whose structure differs from the training distribution — limiting out-of-the-box generality.

Akella and Narayanam [2] proposed QUIS, a question-guided insight generation system for automated data exploration developed at IBM Research. QUIS lets a user pose natural-language questions that are translated into exploration actions. While this improves control over what

is explored, it still presumes the user knows which questions are worth asking, which limits its usefulness for a beginner encountering an unfamiliar dataset for the first time.

C. LLM- and NLP-Driven EDA and Insight Generation

Zhang et al. [8] presented TiInsight, a SQL-based, cross-domain automated EDA system built around an LLM backend and a natural-language query interface. TiInsight demonstrates that LLMs can translate free-text questions into SQL-driven exploration across domains, but the system depends on a SQL execution layer and a GUI shell, and cross-domain generalization is reported as still maturing rather than fully solved.

Dibia [9] introduced LIDA, an LLM-and-image-generation pipeline (from Microsoft Research) that summarizes a dataset in natural language, enumerates visualization goals, and generates and executes chart-producing code. LIDA is one of the more complete systems reviewed in that it couples a natural-language data summary with visualization generation, but its own documentation notes that it performs best on datasets with a small number of columns due to LLM context-length limits, and that executing LLM-generated code introduces a security consideration that must be sandboxed.

Ma et al. [11] presented InsightPilot, an LLM-empowered automated data exploration system demonstrated at EMNLP 2023, which chains a sequence of exploration actions guided by an LLM to surface insights. As a system demonstration, InsightPilot illustrates the feasibility of LLM-guided exploration but is evaluated at demo scale, and insight quality is sensitive to prompt design; the paper does not emphasize an exportable, shareable report as a first-class output.

Hollmann et al. [12] proposed CAAFE, a context-aware automated feature engineering method that uses an LLM to reason about a dataset's semantics and iteratively propose new features, evaluated at NeurIPS 2023. CAAFE demonstrates that an LLM can meaningfully reason about tabular semantics beyond raw statistics, but its scope is feature engineering rather than full EDA reporting, and each iteration incurs an LLM API call, which has a recurring cost implication for large or frequently re-analyzed datasets.

D. Data Profiling Tools in Practice

Beyond the academic literature, ydata-profiling (formerly pandas-profiling) [10] is a widely adopted open-source Python library that generates a single-file HTML profiling report — summary statistics, missing-value counts, correlations and distribution plots — from a pandas or Spark DataFrame with one line of code. It is the closest existing tool to a one-click profiling report, and is frequently used as a baseline in academic evaluations. However, its output is purely descriptive: it does not generate natural-language interpretation of what the statistics imply, and independent evaluations note that report-generation time grows substantially on large or high-cardinality datasets.

IV. COMPARATIVE SUMMARY AND COMMON GAPS

Table I consolidates the methodology and principal limitation of each reviewed work (reproduced below the two-column text for full-width readability). Reviewing the table as a whole, five recurring gaps can be identified:

- 1) Lack of natural-language interpretation. Most rule-based tools ([4], [7], [1], [10]) stop at numbers and charts, leaving interpretation entirely to the user.
- 2) Heavy or narrow infrastructure dependence. Several systems require specific execution environments — a Hadoop/Spark cluster [3], a SQL backend [8], or the R ecosystem [4], [7] — limiting portability for a typical single-analyst workflow.
- 3) Scalability limits. LLM-based tools in particular report degraded performance or accuracy on wide or large datasets due to context-length constraints [9], and descriptive profilers slow down on large tables [10].
- 4) Fragmented tooling. Profiling, visualization, insight generation and feature engineering are addressed by separate, non-integrated tools [1], [9], [12], so a complete workflow requires stitching multiple systems together.
- 5) Limited accessibility for non-programmers and beginners. Tools that rely on writing package calls [4], [7] or on knowing what question to ask [2] remain a barrier for users without a statistics or programming background.

V. TOWARD AN INTEGRATED SOLUTION: THE AI EDA GENERATOR

The gaps identified in Section IV directly motivate the design of our ongoing project, the AI EDA Generator. The system is structured as a single pipeline — ingestion, automated statistical profiling, an AI reasoning layer, and report generation — so that profiling, visualization and interpretation are produced from one upload rather than several disconnected tools, addressing gap 4. A user uploads a CSV, Excel or SQL-connected dataset through a web interface; the system infers schema and types, computes missing-value, duplicate and outlier statistics, and generates standard plots (histograms, box plots, correlation heat-maps) without requiring the user to write any code, addressing gap 5. Unlike the R-specific or cluster-dependent tools in Table I, the pipeline is built on a lightweight Python/FastAPI backend with pandas and NumPy for profiling, which avoids the deployment overhead reported for distributed anomaly-detection frameworks [3], addressing gap 2.

The distinguishing component of the system is an LLM-based reasoning layer that reads the computed statistics — skewness, correlation strength, outlier counts — and drafts a short, plain-language summary of what they imply for the dataset (for example, flagging a strong correlation between two variables and suggesting a possible next step), directly addressing gap 1, the absence of interpretation identified across the rule-based tools reviewed. To mitigate the scalability and cost concerns raised for existing LLM-based

systems [9], [12] (gap 3), the reasoning layer is scoped to operate on pre-aggregated statistics rather than raw records, which keeps the input compact regardless of dataset row count, though very wide datasets with a large number of columns remain a shared limitation with LIDA [9] and are noted as an area for further optimization. In a preliminary evaluation on 25 public datasets across finance, healthcare and retail, the project team's own testing indicated a substantial reduction in analysis time relative to manual scripting; this figure is a self-reported, in-progress result rather than an independently validated benchmark, and is included here to indicate the direction of ongoing evaluation rather than as a settled claim.

VI. CONCLUSION AND FUTURE SCOPE

This review examined twelve representative works spanning statistical autoEDA libraries, ML-guided exploration and LLM-based insight and visualization systems. While each generation of tools has progressively reduced the manual effort required for exploratory data analysis, no single reviewed system offers a complete, lightweight, and interpretable pipeline accessible to non-programmers. The recurring gaps — missing natural-language interpretation, infrastructure dependence, scalability limits, fragmented tooling, and limited beginner accessibility — form a concrete set of design requirements. Our AI EDA Generator project targets these requirements directly by combining automated profiling with an LLM reasoning layer inside a single low-code pipeline. Future work will extend the system toward streaming and database data sources, automated (not just detected) data-cleaning suggestions, tighter integration with downstream ML pipelines, and support for unstructured (text and image) data, in line with directions also noted as open in the LLM-based systems reviewed here [8], [9], [12].

TABLE I
COMPARATIVE SUMMARY OF REVIEWED AUTOMATED/AI-ASSISTED EDA APPROACHES

Ref.	System / Study	Core Approach	Key Limitation
[4]	R AutoEDA landscape survey	Reviews 15 R packages for profiling/visualization	No cross-package interpretation; user must judge significance manually
[7]	SmartEDA (R)	One-call descriptive statistics and plots	Raw output only, no narrative insight; R-specific
[1]	Autoviz	Automated chart generation for classification prep	Visualization only; no profiling or textual explanation
[3]	Hadoop/Spark anomaly framework	Distributed statistical anomaly/outlier detection	Requires big-data cluster infrastructure
[5]	EDA role survey	Catalogues EDA tools and impact of skipping EDA	Descriptive survey; no automation architecture proposed
[6]	ML-based EDA automation overview	Recommenders, active learning, RL agents for exploration	Interestingness models need per-domain training; brittle to new data
[2]	QUIS	Question-guided automated data exploration	Assumes user already knows what to ask
[8]	TiInsight	SQL + LLM cross-domain EDA with NL interface	Needs SQL back-end/GUI; cross-domain generality still maturing
[9]	LIDA	LLM + image-generation pipeline for charts/infographics	Degrades on wide datasets; executes LLM-generated code
[11]	InsightPilot	LLM-chained exploration actions for insight discovery	Demo-scale; prompt-sensitive; no first-class exportable report
[12]	CAAFE	LLM-guided context-aware feature engineering	Scoped to feature engineering, not full EDA; recurring API cost
[10]	ydata-profiling	One-line descriptive HTML profiling report	Purely descriptive; slows markedly on large datasets

REFERENCES

- [1] “Exploratory Data Analysis using Autoviz for Machine Learning Classification Problem,” in Proc. IEEE Conf., 2024.
- [2] A. Akella and K. Narayanam, “QUIS: Question-guided Insights Generation for Automated Exploratory Data Analysis,” arXiv preprint arXiv:2410.10270, 2024.
- [3] “Automating anomaly detection for exploratory data analytics,” in Proc. IEEE Conf., 2018.
- [4] M. Staniak and P. Biecek, “The Landscape of R Packages for Automated Exploratory Data Analysis,” arXiv preprint arXiv:1904.02101, 2019.
- [5] “Role of Exploratory Data Analysis in Data Science,” in Proc. IEEE Conf., 2021.
- [6] “Automating Exploratory Data Analysis via Machine Learning: An Overview,” in Proc. 2020 ACM SIGMOD Int. Conf. Management of Data, 2020, pp. 2617–2622.
- [7] S. Putatunda, K. Rama, and D. U. Ravi Kondapalli, “SmartEDA: An R Package for Automated Exploratory Data Analysis,” J. Statistical Software, arXiv preprint arXiv:1903.04754, 2019.
- [8] D. Zhang, L. Tang, and Q. Liu, “TiInsight: A SQL-based Automated Exploratory Data Analysis System through Large Language Models,” arXiv preprint arXiv:2601.09404, 2026.
- [9] V. Dibia, “LIDA: A Tool for Automatic Generation of Grammar-Agnostic Visualizations and Infographics using Large Language Models,” in Proc. 61st Annual Meeting of the Association for Computational Linguistics (ACL), System Demonstrations, 2023, pp. 113–126.
- [10] YData, “ydata-profiling: Automated Data Profiling and Quality Checks (formerly pandas-profiling),” [Online]. Available: <https://ydata.ai>. Accessed: 2026.
- [11] P. Ma, R. Ding, S. Wang, S. Han, and D. Zhang, “InsightPilot: An LLM-Empowered Automated Data Exploration System,” in Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations, 2023, pp. 346–352.
- [12] N. Hollmann, S. Müller, and F. Hutter, “Large Language Models for Automated Data Science: Introducing CAAFE for Context-Aware Automated Feature Engineering,” in Proc. 37th Conf. Neural Information Processing Systems (NeurIPS), 2023, pp. 44753–44775.