Universiteit Leiden
The Netherlands

Data Science and AI

Integrating Subgroup Discovery with Conversational AI for Enhanced Business Intelligence

Bilal Mohamed - s3611744

Supervisors:
Dr. Arno Knobbe Zhaochun Ren

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

29/07/2025

Abstract

Business intelligence systems today force organizations to choose between simple tools that all employees understand, or powerful analytics requiring data science expertise. This thesis eliminates this trade-off by showing how conversational AI makes enables finding advanced patterns through natural language interaction.

The study combines pySubDisc's advanced subgroup discovery algorithms with OpenAI's GPT-4o. This enables users to ask business questions like "What customer segments drive our highest profits?" and obtain statistically sound, useful answers. The system automatically takes care of complicated algorithmic configuration and converts results into interpretable business explanations.

A full evaluation demonstrates that the statistics generated are accurate and that advanced analytics can be accessed in natural language. User testing indicates that users can use conversational interfaces to explore data effectively, however, find difficulty in extensive analytical conversations with current technology.

The results show that conversational analytics is a step forward in business intelligence, allowing companies to make complex pattern discovery available to its users while keeping the statistical integrity that is necessary for making reliable decisions. Conversational interfaces don't replace existing tools; instead, they make them more accessible and change the way in which organizations use data to get insights.

This study shows that the future lies not in choosing between accessibility and sophistication. Instead, it means having systems delivering both through combining artificial intelligence with established statistical methods.

Contents

1 Introduction

1.1 Problem Context and Research Motivation

Academic Context Conventional business intelligence tools create a gap between advanced analytical functionalities and user accessibility. Modern systems require expertise in data querying, statistical analysis, and dashboard management, which not all employees in a business possess. This level of complexity constrains the efficiency of BI systems in supporting strategic decision making, especially in scenarios in which domain experts need to understand complex patterns without being data science experts. The emergence of explainable AI (XAI) signifies greater awareness about model transparency and reliability in data-driven systems [DVK17]. Subgroup discovery (SD) is a type of supervised data mining that can provide straightforward and interpretable rule-based results and actionable insights in structured datasets [Atz15]. However, SD is not widely used in business intelligence tools due to its complexity of integration and interpretation for non-technical users.

Research Problem Subgroup discovery has proven successful in areas like fraud detection, marketing analytics, and medical diagnostics, however, it is still not widely used as a business intelligence tool. The main issue is accessibility, configuring SD runs requires extensive parameter adjustment (e.g., support thresholds and quality measures), preprocessing and validation, which can be challenging for non-technical users. Large language models (LLMs) like GPT-4o have proven effective in natural language understanding (NLU), natural language processing (NLP), and natural language generation (NLG) [Ope23]. This indicates a significant opportunity to utilize LLMs as natural language interfaces to make SD-based analysis understandable to non-technical users.

Innovation Opportunity Integrating an established subgroup discovery algorithm with conversational AI presents an opportunity to build conversational business intelligence systems that maintain statistical integrity while improving interpretability and user interaction. This research proposes a real time interactive system which integrates PySubDisc, an advanced SD algorithm, with OpenAI's GPT-4o. This allows non-technical business users to ask questions like "What influences my sales at Location X?" and obtain interpretable, statistically supported answers.

1.2 Research Gap Analysis and Innovation Positioning

Literature Gap Analysis Previous studies have investigated the use of natural language interfaces to search databases [LJ14] and the use of explainable AI in business settings [A+20]. Currently, there is no systematic integration of data mining algorithms such as SD with large language model-powered conversational interfaces. Current BI tools, like Power BI and Tableau Ask Data, allow users to ask questions in plain language to get data, however, lack support for multi-step analytical processes like SD. Investigations into XAI also frequently focus on model explanation (such as feature attribution in classifiers) instead of statistical pattern extraction.

Technical Innovation This project introduces three AI-enhanced components:

- **Intelligent parameter selection**: GPT-4o infers SD configurations such as optimal quality measures and numeric strategy based on user queries; fallback methods apply validated defaults when confidence is low.
- **Dynamic target extraction**: GPT-4o infers the optimal SD target for algorithm runs from queries and extracts targets from the dataset.
- **Context-aware business explanation generation**: Business strategy and advice generated by LLM based on subgroup discovery results.

Positioning Statement. This study specifically tackles the integration challenges by systematic engineering and prompting of AI-enhanced statistical analysis. This study creates new methodological frameworks for conversational business intelligence, validated through technical benchmarking and user-centered assessment.

1.3 Research Questions and Objectives

Primary Research Question:

> *How does the integration of subgroup discovery with conversational AI enhance the accessibility and actionability of business intelligence for non-technical users?*

Supporting Research Questions.

- **Technical effectiveness (RQ1):** To what extent does AI-driven parameter selection and natural language generation maintain statistical rigor while improving usability?
- **User experience impact (RQ2):** How does conversational interaction with statistical patterns affect comprehension and decision accuracy?
- **Business value creation (RQ3):** What measurable benefits arise in decision speed, confidence, and outcome quality?

1.4 Research Objectives and Thesis Structure

The primary focus of this research project is to design and test an integrated system using subgroup discovery algorithms and conversational AI to enhance the accessibility and effectiveness of business intelligence. Specifically, this thesis aims to:

- To design and implement a novel architecture that combines `pysubdisc`-based subgroup discovery with OpenAI's LLM capabilities to create an interactive business intelligence platform.
- To develop effective methods for translating complex subgroup patterns into interpretable and decision oriented natural language explanations suitable for business decision making.
- Conduct comprehensive evaluation of the system's effectiveness through both technical performance metrics and user experience studies.

1.5 Thesis Structure and Contribution Summary

This thesis develops a method to combine subgroup discovery algorithms with conversational AI. It achieved this by going through nine chapters in a systematic way, from theoretical foundations to practical implementation and comprehensive evaluation.

The theoretical development covers Chapters 2 and 3, which lay the groundwork for the research by analyzing the literature and providing technical background on the foundations of subgroup discovery. Chapters 4 and 5 discuss the main innovation: a new four-layer architecture and systematic prompt engineering method that allows users to use conversational AI to obtain advanced analytics. Chapter 6 addresses the evaluation framework, which is a full three part assessment method that balances technical validation with usability for users. Chapters 7–9 demonstrate the results, discuss implications for business intelligence practice, and synthesize the contributions to the field.

Research Contributions:
This work makes conversational business intelligence through four important new features. The "theoretical contribution" shows that systematic AI integration eliminates the traditional trade-off between accessibility and rigor in advanced analytics. The "methodological innovation" creates a framework for prompt engineering that can be used to make complicated algorithms simpler to interpret. The "practical achievement" discusses the working example of conversational subgroup discovery for business use. The evaluation framework enables evaluation of the conversational analytics system that takes into account both statistical validation and user experience.

These contributions address the challenge of making advanced analytical tools more accessible to business users. They show that conversational analytics is a promising way to make complex data mining techniques interpretable to non-technical users while maintaining the integrity of the analysis.

2 Background and Related Work

2.1 Data-Driven Decisions and Interpretable AI

In the modern competitive business landscape, organizations are increasingly aware of the vast potential presented by data-driven decision making for optimize operations and performance [Keb22]. The Fourth Industrial Revolution has thoroughly transformed business operations as companies establish business intelligence as a key component of data-driven decision-making processes, leveraging data analytics across multiple organizational aspects, particularly artificial intelligence and big data analytics. A study highlighting the benefits of becoming a data-driven organization demonstrated that data-driven organizations are 23 times more likely to acquire customers, 6 times more likely to retain customers, and 19 times more likely to have profits above their competition [BV22]. Organizations involved in data-driven decision-making also experience 5% higher productivity and 6% higher profitability than their competitors [Ana23].

The business intelligence market has grown rapidly and is estimated to be worth $56.9 billion by 2032, with an annual growth rate of 7.2% [Res25]. However, approaches to data analysis can undermine the interpretability and accessibility needed to facilitate effective decision making. The implementation of machine learning and data mining across industries is creating new needs to manage and maintain the learned models. This led to the emergence of explainable artificial intelligence (XAI) as a research field, highlighting the importance for stakeholders to interpret machine learning models [AFKS24].

2.2 Subgroup Discovery: Extracting Patterns from Business Data

Subgroup discovery is a practical data mining algorithm that addresses the challenge of discovering interesting and statistically significant patterns within large datasets [Her11]. While classic classification methodology is focused on maximizing the accuracy of the classification, subgroup discovery merges descriptive indications with predictive reasoning in order to find interesting distributions defined by a target variable of interest. The goal of subgroup discovery is to find subgroups in a population that exhibit statistically interesting patterns, which is defined by how large of a subgroup can be found, and displays the most interesting distributions in relation to the target variable.

This approach has been applied to many business applications such as marketing audits, customer clustering , fraud alerts, and workflow tuning [Atz15]. Subgroup discovery has an advantage in that it provides simple, interpretable results and data relations which are useful for business intelligence problems when understanding the "why" behind a specific patterns is equally important as just finding the patterns as they arise.

2.3 Accessibility Challenge: Bridging the Gap Between Insights and Business Users

Although techniques such as subgroup discovery offer great potential for enhancing business intelligence initiatives [CE11], there remains a disconnect between technical insight and interpreting that insight for non-technical business stakeholders. Current BI technologies require a high level of technical ability to generate and review a dashboard, and this can limit the flow and accessibility of data-driven insights within an organization. Explaining and interpreting the complex implications of a data-driven insight to non-technical stakeholders presents a natural tension to the business, as these are the individuals with significant responsibility in the decision-making process, even if they do not have the technical expertise to demonstrate much depth of knowledge [Amj23].

Evidence suggests that the disconnect between analytic skill and user data accessibility limits how often advanced data-driven insights are utilized and applied to daily operations. Non-technical team members frequently demonstrate reluctance towards data mining findings relative to the technical complexity of data representations [Unk25]. This issue is highlighted in the business-intelligence context, as the true potential of deeper analyses will remain underutilized, understood only when findings are translated for non-technical decision-makers.

The use of static dashboards and reports reveals this issue, as they are difficult to manipulate and cannot create the dynamics of data exploration that business users need to analyze their problem [Dah24]. Business solutions need to bridge this gap and transform complex analytical data into easily interpretable actionable data while maintaining the complexity and structure.

2.4 Large Language Models: Data Insights into Natural Language

The implementation of Large Language Models (LLMs) represents a technological advancement in the field of natural language processing as it can bridge the gap between complex data analysis and interpretable explanations. LLMs are deep learning algorithms trained on large corpora of data that perform a variety of natural language processing tasks such as text generation, translation and summarization, while demonstrating the ability to understand contexts and produce human-like responses [Ela25].

Their true potential lies in their advantageous ability to express complex patterns and insights in natural language that are easily interpretable. This reduces the complexity of information presentation for non-technical users. The research suggested that LLMs can generate natural language explanations that could be used to support the reason behind analytical insights, and in doing so make complex data insightful, while reducing the need for the technical requirements [MH25].

In business contexts, LLMs are becoming increasingly effective in automating both the process of interacting with data and the decision making itself. These companies are finding that conversational AI which is built using LLMs can have a breadth of applications including: automatically translating user inputs into data insights that can be analyzed by simply providing an intuitive interface for data exploration [BHH+24]. These applications demonstrate the potential that LLMs have in

connecting complex analytical systems to non-technical users, ultimately allowing for a wider net of data and insights.

3 Technical Background

3.1 Subgroup Discovery as a Data Mining Paradigm

Subgroup discovery is a distinctive data mining paradigm that originates from a need to identify patterns in complex datasets that is interpretable and maintains analytical rigor [Klö92]. This paradigm differs from traditional machine learning methods as it focuses on identifying locally interesting subgroups rather than optimizing global predictive performance. Klösgen's foundational work established the theoretical framework for finding subgroups that are interesting with respect to a target variable. This work emphasizes the importance of having interpretable pattern descriptions that support users in making decisions.

The paradigm's theoretical development was advanced further through systematic research that identified the need for descriptive pattern mining approaches that are capable of connecting purely descriptive statistics and predictive modeling [Atz15]. This development made subgroup discovery a principled approach of identifying local patterns that demonstrate a mixture of accuracy and reliability in their target behavior. Rather than looking for patterns that achieve the highest global classification performance, the paradigm looks for rules that are interpretable and have high target concentration, sufficient coverage for practical relevance, and a substantial deviation from baseline population behavior.

The philosophy sets subgroup discovery apart from other data mining approaches by ensuring that the patterns discovered can be expressed as interpretable combinations of simple attribute conditions. This feature makes the paradigm especially useful for business intelligence applications, where users need accurate insights and clear explanations of the factors that influence business outcomes. Each discovered subgroup naturally leads to business rules that are actionable and enable high comprehension and strategic implementation.

3.2 Fundamental Concepts and Mathematical Foundation

Before explaining the technical details of subgroup discovery algorithms, it's important to understand the core concepts that form the foundation of this paradigm. The mathematical framework operates on structured datasets, where patterns can be identified and evaluated according to set criteria.

The basic data structure is a dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. Here, $x \in X$ is the feature space that contains attribute values, and $y \in Y$ is the target variable of interest. In this context, subgroup discovery identifies meaningful subsets $S \subseteq D$ that show interesting deviations from the expected target distributions.

The pattern representation utilizes subgroup descriptions in the form of logical conjunctions. A subgroup description (sd) consists of a set of attribute conditions $sd = c_1 \wedge c_2 \wedge \ldots \wedge c_k$, where each condition c_i constrains specific attribute values. The logical structure enables understanding for both the user and computer processing. Categorical attributes produce conditions such as "attribute

= value," and numerical attributes produce threshold-based constraints using relational operators.

The subgroup G that is produced the description sd includes all the instances in the dataset that meet the conditions: $G = \{(x, y) \in D | sd(x) = \text{true}\}$. This formal definition is the basis for all future procedures for evaluating quality and patterns.

3.3 Core Evaluation Metrics

In order to assess the quality of a subgroup, several key metrics that measure different aspects of pattern significance and reliability need to be observed. These numbers enable comparison and ranking between discovered patterns based on usefulness for analysis and business value.

Coverage is the most basic metric. It measures the size of a subgroup through a simple count $n = |G|$. This metric measures how many dataset instances meet the subgroup conditions, which represents usefulness and applicability of the patterns discovered. Coverage is a fundamental building block for more advanced output metrics while offering insight into pattern breadth.

Target Share represents the amount of target instances in a subgroup. It is defined as $t_P = |\{(x, y) \in G | y = \text{target\_value}\}|/|G|$. This ratio represents the strength the target phenomenon is in the subgroup, which is the basis for assessing the accuracy of the pattern. The baseline target rate t_0 represents the expected target proportion in the dataset. It is used as a reference point for measuring deviation and understanding the significance of discovered patterns.

Quality measures the overall interestingness of a discovered subgroup through a score that is computed by combining coverage and target deviation information. This metric represents the value calculated by the selected quality measure (such as WRAcc, Lift, Binomial, or Cortana Quality) and provides the primary mechanism for ranking and comparing discovered patterns. Higher quality scores generally indicate more interesting and valuable patterns for business applications.

These three core metrics provide comprehensive characterization of each discovered subgroup, enabling technical evaluation and business interpretation of pattern significance. The output framework ensures consistent interpretation across different analytical contexts and enables smooth integration with downstream business intelligence processes.

3.4 Quality Measure Framework and Strategic Application

The evaluation of subgroup quality has become a complex mathematical framework that unifies different assessment approaches in a single theoretical framework. This unification demonstrates the basic links between different quality measures and enables selection of the most optimal quality measure based on specific business needs and analytical goals.

The theoretical basis is the generalized quality measure formulation $q_S^\alpha(P) = n^\alpha \cdot (t_P - t_0)$, where the parameter $\alpha \in [0, 1]$ controls the relative influence of subgroup size versus pure target devia-

tion. This parameterization demonstrates that each quality measures have different points on a continuous spectrum of size-accuracy trade-offs. This enables selection based on analytical priorities.

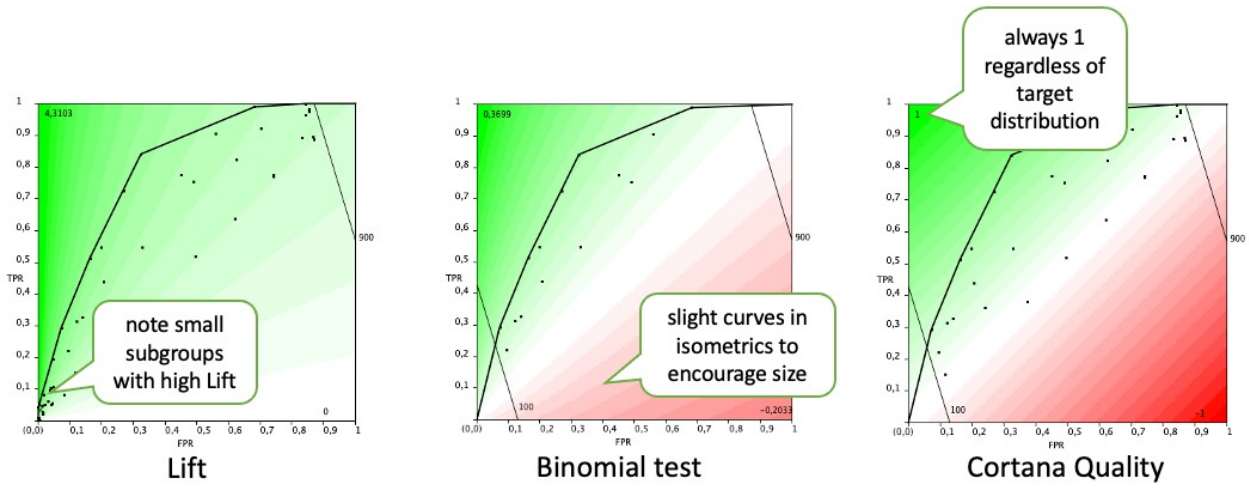


Figure 1: Quality measure isometric curves showing different optimization behaviors for Lift, Binomial test, and Cortana Quality

Lift Quality ($\alpha = 0$) removes the effect of subgroup size by using the formula $\text{Lift}(G) = t_P/t_0$, which assesses the target concentration compared to the baseline. This multiplicative ratio is simple to understand for users. Values over 1.0 mean that the target is getting more concentrated, and higher values mean that the concentration effects are stronger. As shown in Figure 1, Lift exhibits straight isometric curves that note small subgroups with high Lift values.

- **Strategic Use:** Lift is best for business situations that require precision, like targeted marketing campaigns where finding the customer segments with high conversion takes precedence over the size of the segments, fraud detection applications that require identifying suspicious patterns with high accuracy, or premium product targeting where small, high-value customer groups require focused attention.

Binomial Quality ($\alpha = 1/2$) uses square-root size weighting to find a balance between subgroup size and target share. This is done by using $q_S^{\alpha}(P) = n^{1/2} \cdot (t_p - t_0)$. The square root function grows more slowly as its argument increases. This means that this measure doesn't necessarily favor very large subgroups in the way other quality measures.

- **Behavior of Optimization:** This measure is suitable for finding medium sized subgroups that still show interesting subpopulations of the data. The curved isometrics enables subgroups on the left side of ROC space, with the top-left corner of ROC space scoring the highest.

- **Strategic Use:** Binomial quality is best for situations where subgroups of moderate size are preferred over subgroups that are large or too small. This measure is suitable in applications requiring balanced discrimination between pattern strength and subgroup coverage, such as medical research applications, risk management situations where confidence guarantees are required, or business contexts where both precision and reasonable sample sizes matter.

- **Computational Characteristics:** The measure provides robust pattern detection while keeping sensitive to important subpopulations. However, this measure isn't favorable for smaller datasets due to the mathematical foundation becoming less reliable. The curved isometric behavior suitable for finding different types of subgroups in different regions of the search space.

Weighted Relative Accuracy ($\alpha = 1$) uses full linear size weighting with $\text{WRAcc}(G) = (|G|/|D|) \times (t_P - t_0)$. This ensures that larger subgroups obtain higher quality scores when demonstrating equivalent target deviation. This aligns with the practical rule that patterns that affect more cases provide more business value.

- **Strategic Use:** WRAcc is the best choice for general business intelligence applications where pattern strength and operational scope are equally important for strategic value, operational improvement projects that is required to be sufficiently significant to have an impact, customer segmentation projects that require balancing accuracy and subgroup size for practical marketing implementation, or resource allocation decisions where the number of affected customers directly affects business outcomes.

Cortana Quality ($\alpha = 1$) maintains order-equivalence with WRAcc while providing normalized quality scores that are interpretable in the range $[-1, 1]$ through the formula $q_{cq}(S) = (t_P - t_0) \cdot |G|/|D|_{\max}$, where $|D|_{\max}$ is the theoretical maximum subgroup size. This normalization simplifies comparison across the datasets with different baseline characteristics while maintaining the linear size weighting philosophy. As indicated in Figure 1, Cortana Quality is always 1 regardless of target distribution.

- **Strategic Use:** Cortana Quality is useful for comparative business intelligence applications that require observing patterns across multiple datasets, time periods, or market segments; dashboard applications that simplify comprehension of scores regardless of underlying data; or benchmark analysis that compares subgroup quality across different business units or geographic areas. Cortana Quality is particularly useful for executive reporting due to the score range that it offers which simplifies understanding across different analytical contexts.

3.5 Numerical Attribute Processing Strategies and Business Context

The handling of continuous numerical attributes is a crucial algorithmic choice that impacts both the quality of the analysis and interpretability. The strategic choice between processing methods must be in line with business goals to obtain optimal results.

The **NUMERIC_BINS** strategy uses equal-width discretization to divide continuous attribute ranges into intervals of equal width. This method enables understanding of threshold boundaries at regular intervals. It does this by making split points of equal distances between the minimum and maximum values of the numeric variable in each subgroup.

The "number of bins" parameter (nbins) is used by the strategy to determine the number equal-width intervals are created. If the numeric variable has fewer than nbins values, all of its unique values are used as splitting points. The result list and the candidate queue for the next depth both obtain all of the possible splits.

Computational Efficiency: By limiting the search space to nbins split points instead of checking every unique value in the dataset, NUMERIC_BINS requires less computation power This makes it suitable for working with large datasets where speed is more important than thorough optimization.

Strategic Use: NUMERIC_BINS is helpful in business situations where users value understanding, working with other systems, and speed. The equal-width intervals simplifies for field staff, operational teams, and non-technical stakeholders to understand by creating round-number boundaries. However, this strategy may sacrifice analytical accuracy and may miss optimal discrimination points that fall between the set intervals.

NUMERIC_BEST Strategy uses information-theoretic criteria, to identify cut-points that maxmize discriminative power by optimizing quality-based thresholds. This method assesses all the possible split points in the attribute range and selects the best informative partitions based on the quality measure being used.

The subgroup only obtains the best split point for each candidate subgroup. After that, this subgroup is added to the list of results and the queue of candidates for the next depth.

Computational Intensity: NUMERIC_BEST requires more computing power as it assesses every unique value in the dataset as a possible split point. For large datasets, this method can be very expensive in terms of computational power, but it guarantees mathematically optimal discrimination.

Strategic Use: NUMERIC_BEST works best for algorithmic trading systems, automated decision systems, and machine learning pipelines where direct performance effects and enough computing power are available. The strategy should be selected when discovered patterns will be implemented through automated systems that can handle exact numerical thresholds without requiring human interpretation or manual implementation.

3.6 Search Depth Configuration and Complexity Management

The complexity of discovered subgroup patterns depend on search depth configuration. This is due to the fact that search depth limits the maximum number of attribute conditions that can be combined using logical conjunction. This basic selection has a direct effect on interpretability of results and computational requirements.

Depth 1 Configuration limits subgroup descriptions to single attribute conditions, creating uni-variate patterns like "Gender = Female" or "Age $\geq$ 65" that only evaluate direct relationships between attributes and targets. **Strategic Application:** The Depth 1 configuration is best for the early stages of exploratory analysis, where understanding takes precedence over finding all the subgroups and for executive summary applications which require understanding of patterns immediately. This configuration proves useful for finding the main factors that affect business outcomes before advancing to complicated interaction analysis. However, it is incapable of identifying relationships between multiple factors that might represent most important business insights.

Depth 2 Configuration enables bi-variate subgroup descriptions through combining two attribute conditions with logical conjunction. For example, "Gender = Female AND Age $\geq$ 65" or "Product_Category = Electronics AND Customer_Tier = Premium." **Strategic Application:** Depth 2 configuration is necessary for full business intelligence applications that require interaction effect analysis, customer segmentation projects require assessing combinations of behavioral patterns, or analytical environments where users can understand complex pattern relationships. This system should be used when there is sufficient data to reliably detect interactions.

3.7 Algorithmic Development and Implementation Landscape

The algorithmic foundation of subgroup discovery has evolved over decades of research innovation, building from early theoretical work to sophisticated modern implementations. Research established that systematic subgroup identification was feasible and valuable for systematic subgroup identification [Klö92]. This early work set up the frameworks that later algorithmic advances would build on.

CN2-SD was one of the first algorithms to combine subgroup discovery goals with established rule learning methods. This integration created the foundation for later algorithmic improvements that would enhance both the algorithm quality and computational speed. Beam Search strategies grew into the most popular approach of exploration, using best-first search with adjustable beam widths to efficiently compute through an exponentially large space of possible subgroup descriptions.

SD-Map represents an advancement in computational efficiency as it uses optimized data structures and smart pruning strategies to increase computational speed pattern discovery in large datasets. The pysubgroup package is a full Python implementation that has become popular within the research community as its API is simple to use and it integrates with other Python data science tools.

3.8 SubDisc and pySubDisc: Implementation and Research Foundation

SubDisc previously known as Cortana is a sophisticated subgroup discovery system that has been in development for more than ten years at Leiden University. This mature platform has served as the foundation for extensive research in subgroup discovery algorithms demonstrating its robustness and reliability through over a decade of continuous refinement and optimization.

The core SubDisc system is a comprehensive Java based platform for subgroup discovery across various target types, including nominal classification, numeric regression, and multi-target scenarios [MK11]. Its architecture incorporates advanced pruning strategies and optimized data structures that enable efficient processing of large datasets while maintaining analytical rigor [KCFS08]. .

pySubDisc represents a recent addition to this established ecosystem, serving as a Python wrapper for the mature Java subgroup discovery tool SubDisc [Pal23]. The creation of pySubDisc reflects the extensive research experience gained from both theoretical subgroup discovery developments and real-world implementation challenges encountered over more than ten years of system evolution. This Python interface bridges the gap between the mature Java-based SubDisc engine and the Python data science ecosystem, making the powerful subgroup discovery capabilities accessible to Python users working with pandas DataFrames.

The pySubDisc wrapper inherits the computational advantages of its underlying Java engine, including the ability to parallelize tasks and optimize algorithmic implementations. This enables processing of large datasets quickly while maintaining high quality solutions. pySubDisc provides a comprehensive set of analytical tools that handle both classification and regression tasks with a wide range of target variable types. In classification situations, the engine handles binary and multi-class nominal targets using optimal algorithms for predicting categorical outcomes. The regression function handles continuous numerical targets through quality measures and evaluation criteria specifically adjusted for numeric target variables, meeting a wide range of analytical needs across numerous domains and business applications.

3.9 System Implementation: singleNominalTarget Configuration

In this research system, pySubDisc is the main analytical engine, and it is used in a particular manner that takes advantage of the engine's singleNominalTarget functionality. This implementation choice is based on the business intelligence requirements for the conversational system, takes precedence on clear categorical target analysis as opposed to all of pySubDisc's analytical features. The singleNominalTarget configuration provides the best performance for both binary and multi-class classification situations while enabling parameter flexibility, which is required for smart configuration adaptation. This focused approach makes the system efficient and simplifies to explain identified subgroups to business users in natural language.

Integration Architecture: The system integrates pySubDisc through a wrapper that handles parameter configuration, coordinating execution, and returning results using AI. Using the framework from Section 4.4, AI-driven query classification method maps conversational user intents to the optimal parameter configuration.

The Result Interpretation Pipeline: outputs from the algorithm are transformed into business intelligence insights for the discovered subgroups. The pipeline puts the results of pySubDisc into structured forms that include information about the target, conditions for the subgroup, output metrics, and business context metadata. These structured outputs are used as input for the AI model prompts that create natural language explanations that are interpretable for non-technical users.

This technical foundation makes the thesis's main contribution possible: closing the gap between complex pattern discovery algorithms and useful business intelligence applications by using conversational interfaces.

4 System Design and Architecture

This section focuses on the architectural framework that the system is built upon and facilitates the integration between pySubDisc and conversational AI.

4.1 Requirements Engineering and Design Constraints

4.1.1 Functional Requirements Derivation from Research Questions

The functional requirements of this system focus on the core research question: "How does the integration of subgroup discovery with conversational AI enhance the accessibility of business intelligence for non-technical users?" This research entails specific requirements to demonstrate a solution that bridges the gap between data analysis and user-friendly interaction.

Primary Functional Requirements:

1. **Conversational Query Processing:** Users can express analytical queries in natural language without understanding algorithmic parameters or statistical concepts. The goal of this study is to simplify access and understanding of data analysis for non-technical users.

2. **Automated Subgroup Discovery Runs:** The system must be able to automatically configure and run subgroup discovery according to the user query and target. This eliminates the technical expertise needed for configuration and implementation of a subgroup discovery run. This aligns with the research's objective of making data analytics accessible to users without expert knowledge.

3. **Business-Contextual Result Translation:** The system must transform subgroup discovery results into insightful business-related interpretation and provide strategic recommendations. This supports the study's objective of converting analytical data into actionable insights to support decision making.

4. **Session-Persistent Conversational Context:** The system must maintain conversation state across interactions and can progressively support user queries and conversation flow rather than isolated query-response patterns. This requirement aligns with the study's objective for authentic and natural business interaction.

4.1.2 Non-Functional Requirements with Measurable Criteria

The following non-functional requirements set clear, measurable standards for performance and quality that ensure the system adheres to the research objectives.

1. **Interactive Response Time:** The system should ensure that simple data retrieval queries are handled promptly, and more complex queries requiring subgroup discovery are handled within reasonable time frames. This requirement ensures users have access to insights that aid in decision making in real time.

2. **Algorithm Availability:** The system ensures that algorithms are always available by using backup methods when the main algorithms encounter issues. This requirement pertains to the need for consistent analytical functionality.

3. **Data Integrity Assurance:** The system must maintain statistical accuracy throughout the processing process with zero tolerance for data corruption and AI hallucination. This requirement ensures the accuracy of the results and the trustworthiness of business decisions.

4. **Accessibility Compliance:** The system must allow non-technical users to gain analytical insights without the need for statistical or data expertise. This criterion is directly related to the main goal of the research, which is to improve accessibility of technical information to non-technical business users.

4.1.3 Design Constraints and Trade-off Analysis

The system design operates within several constraints that affected architectural decisions. This requires trade-off analysis to ensure solution validity while focusing on the research objectives.

Technical Limitations:

External API Dependency: The reliance on OpenAI's GPT-4o API creates latency and availability issues that affect the design of the caching and fallback systems. The trade-off analysis favored utilizing natural language understanding from external AI capability over local processing, while acknowledging the risks of dependency.

Limited Resources:

Development Timeline: The timeline for the bachelor's thesis made the architecture less complex by putting established technologies (Flask, HTML/CSS/JavaScript) ahead of new frameworks. Furthermore, the system's functionality is limited as it employs categorical subgroup discovery compared to the full capabilities available in pySubDisc. This trade-off put more emphasis on research validation than on full integration and innovation.

Computational Resources: The performance optimization strategy was limited by the dataset size (3,900 records) and the algorithmic complexity. The strategy focused on choosing optimal parameters instead of using distributed processing architectures.

4.2 Architectural Design Philosophy

4.2.1 Architectural Pattern Selection

The architectural pattern selection process assessed three main approaches: microservices, monolithic, and layered structures. However, the layered structure was favored for this solution.

Reasons for Choosing Layered Architecture:

The system uses a four layer architectural framework that ensures the separation of the system's functionality which is suitable for research validation and the study's parameters. This pattern selection was supported by Fowler's (2002) architectural principles as it aims to reduce the difficulty of combining diverse components (pySubDisc integration, AI components, web interfaces) into a unified system [FRF+02]. The following system layers work together to power the system, as shown in Figure 1:

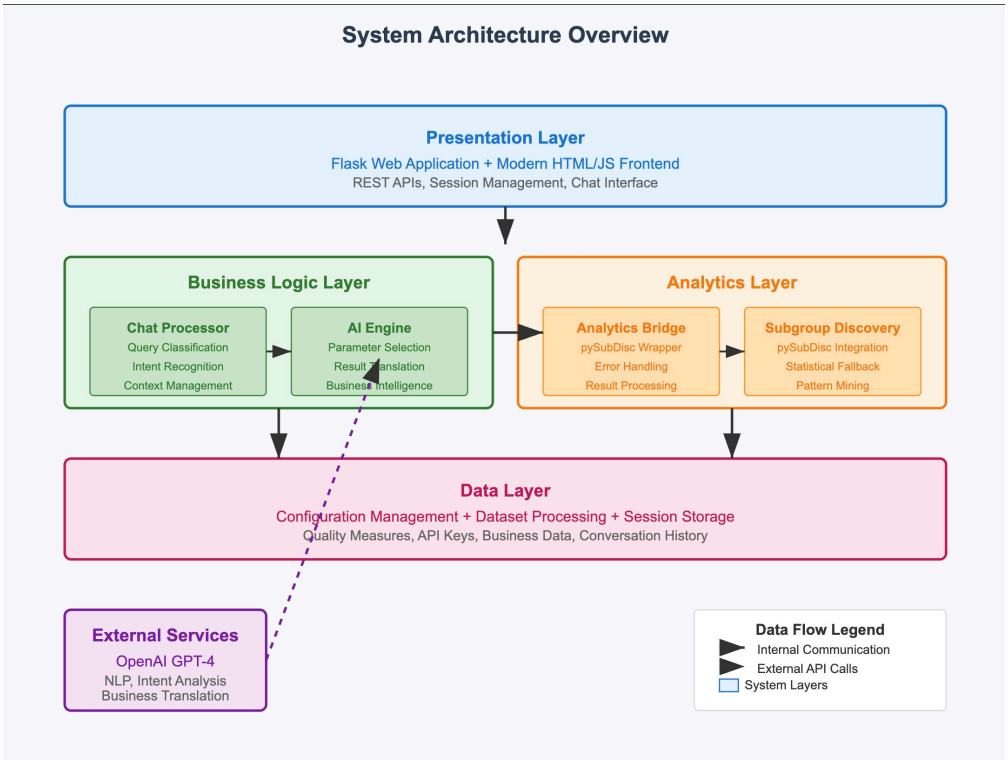


Figure 2: System Architecture Diagram

1. **Presentation Layer:** This layer is responsible for the conversational interface which uses a HTML/JavaScript frontend and Flask API endpoints. This layer separates business logic and user interaction errors, allowing for interface development while preserving the emphasis on conversational efficacy.

2. **Business Logic Layer:** This layer includes system functionality that uses AI components to process queries and generate insightful responses. This layer represents the primary research breakthrough in intelligent query routing and natural language generation for business insights.

3. **Analytics Layer:** This layer bridges the gap between AI and pySubDisc directly. This layer provides the statistical accuracy needed, while hiding the complexity of the algorithm's output from the user.

4. **Data Layer:** The configuration of the system and data management files are utilized in loading and preparing the datasets. This layer ensures data integrity and simplifies access to the dataset.

Analysis of Rejected Alternative Pattern:
The Microservices Architecture was rejected due to it's scope being beyond that of a bachelor's thesis and the constraints of the development time frame. Microservices are more scalable and offer automatic deployment, but this study focuses on new ways to integrate algorithms rather than on building distributed systems. [AY+24]

Monolithic Architecture was not chosen as it could strongly connect AI processing, statistical methods, and the web interface, making it harder to separate and evaluate the different components and its connection to the conversational AI and subgroup results integration[EA22].

4.2.2 Component Responsibility Allocation and Separation of Concerns

System functionality is divided among the components that follow the single responsibility principle. Each component works on a distinct aspect of the system functionality and ensures that the interfaces for integration and testing are clear.

Query Processing Component Responsibilities:
The system's chat processor is responsible for question interpretation, intent classification, and response routing. This component is based on understanding user queries and providing a reliable way for handling conversations. The separation simplifies independent evaluation of the accuracy of query classification and facilitates iterative improvements in natural language understanding ability within the system.

AI Integration Component Responsibilities:
This handles the AI model's understanding of user input and intent, extraction of target from user query, and translation of results from pySubdisc runs. This evaluates the effectiveness of AI-driven business intelligence functions while preserving modular system design.

Analytics Component Responsibilities:
This handles the output of pySubDisc and runs analytics that are specific to the user query and target. This component ensures statistical accuracy while reducing the algorithm's complexity. It supports the study's goal of making analytics available to non-technical users through smart configuration management.

Interface Component Responsibilities:
The Flask application is responsible for HTTP connections, managing sessions, and handling errors. This design allows assessment of user interaction from the interface standpoint without the interference of technical results such as subgroup discovery output.

4.2.3 Scalability and Maintainability Design Decisions

Design decisions on scalability and maintainability balance research objectives with solution practicality, understanding that the main objective is research validation rather than production-level operation.

Vertical Scalability Design: The system focuses on vertical scalability through efficient use of resources instead of horizontal distribution. This decision corresponds with the research's primary focus on efficient algorithm integration.

Configuration-Driven Adaptability: The configuration module ensures research parameters can be changed without altering the code. This method simplifies testing different subgroup discovery parameters, parameter definitions, and performance thresholds while maintaining system stability.

Modular Component Design: System components maintain autonomy through clearly defined interfaces. This ensures that individual components can be modified without affecting the whole system.

Error Isolation and Recovery: Error handling includes fallback methods to restrict failures to certain components of the system while maintaining functionality as a whole. This method ensures that study evaluation is possible and enables easier identification of issue location, which strengthens the validity of research.

4.3 Data Architecture Design

4.3.1 Data Flow Design and Processing Pipeline Architecture

The data flow architecture establishes a processing pipeline that takes raw business data through different phases to generate conversational business intelligence outputs.

Primary Data Flow Sequence:

1. **Data Ingestion and Validation:** Data Loading allows the user to import CSV business data. The design focuses on making data sources flexible to the user and tailoring the solution to business requirements and data.

2. **Business-Centric Preprocessing:** processes raw data by removing customer IDs to eliminate uninsightful subgroups from forming, adding cardinality constraints that ensure analysis efficiency.

3. **Intelligent Query Processing:** User queries go through a multi-stage review by query classification method to process the query appropriately. This structure allows the AI understanding of user query and enables effective routing of queries.

4. **AI-Driven Algorithm Configuration:** The system automatically sets subgroup discovery parameters, based on the query and the business context. The main research breakthrough is the fact that the automation makes expert knowledge on configuration unnecessary, therefore, benefiting non-technical users.

5. **Execution of Statistical Analysis:** The system runs the configured subgroup discovery run and incorporates extensive error handling and fallback protocols. The architecture ensures that the analysis is reliable and simplifies the assessment of various algorithm configurations.

6. **Business Intelligence Translation:** The system converts statistical results from the subgroup discovery into business insights, and it generates explanations that are relevant to the context of the user query.

4.4 AI-Analytics Integration Architecture

4.4.1 Integration Pattern Design for AI + Subgroup Discovery

The AI-analytics integration employs a hybrid approach that combines the Mediator and Strategy patterns. This simplifies the integration between conversational AI and pySubDisc in real time, while keeping them loosely coupled and extensible [GHJV94].

Mediator Pattern Implementation: The system's AI engine acts as an intermediary between user natural language input and the subgroup discovery run requirements. It transforms user queries into technical specifications for the subgroup discovery run without requiring direct integration between components. This framework enables direct assessment of the efficacy of the AI-driven algorithm setup while preserving the capability to interchange various AI functionality or statistical algorithms on its own.

Strategy Pattern for Algorithm Selection: The system employs the Strategy pattern in configurable input metrics for the subgroup discovery run that are dynamically chosen depending on the AI models' assessment of user intent. This is done through defining each of the metrics in AI prompting and the AI recognizes based on the user query which parameters are most appropriate [GHJV94].

4.4.2 API Design for LLM Integration

The design for integrating large language models focuses on dependability, cost-effectiveness, and research validation, all while maintaining the functionality of the LLM for processing user queries.

Cost-Optimized Token Management: The API architecture includes smart token usage optimization through token output restrictions per prompt. This method reduces the risk of over generating information, while preserving response quality.

Context-Aware Request Optimization: API requests include comprehensive context recognition that enables appropriate natural language generation. This design supports research evaluation of conversational effectiveness and quality.

4.4.3 Error Handling and Fallback Mechanism Design

The error handling system is designed to ensure that the system remains functional while maintaining the quality of the user experience and accuracy.

Layered Fallback Strategy: Alternate AI responses, statistical method options, and clarification messages are employed as fallback to the core functionality

Intelligent Error Recovery: In the case of an error, context-sensitive recovery mechanisms are used to respond appropriately to the user's query through different processing paths rather than an error message.

5 Implementation and Technical Innovation

5.1 Business Data Processing Implementation

5.1.1 Direct Data Analysis and Statistical Calculation Engine

The implementation sets up the data processing framework that turns raw business datasets into insightful information. The system processes transactional data that include customer demographics, purchase behaviors, and seasonal patterns using preprocessing workflows that ensure data accuracy and allow simple data retrieval.

The statistical calculation engine uses dynamic aggregation and filtering to handle direct queries that require operations on the dataset values. This enables calculations such as revenue totals and customer segmentation statistics in real time.

5.1.2 Real-Time Query Processing and Response Generation

The query processing framework turns user business queries written in natural language into analytical operations. The system uses contextual analysis to deduce the appropriate approach to process requests. The system can differentiate between data retrieval requests and those that require a subgroup discovery run.

Response generation mechanisms adapt based on the user query and context. When the user requests simple data, it performs statistical calculations and answers the query concisely. When asked a query that requires subgroup discovery, the workflow for running a subgroup discovery run begins. The implementation ensures the consistent quality of responses across different types of queries by using standardized format prompting. The architecture for real-time processing also ensures that the system is responsive.

5.2 AI Prompt Engineering and Subgroup Discovery Training

5.2.1 Core Prompt Engineering Strategies for Algorithm Concepts

The implementation utilizes prompt engineering that turns subgroup discovery into understandable frameworks for the LLM. The training strategy teaches AI systems the details of subgroup discovery and business insights reporting.

The prompt engineering framework introduces algorithmic concepts to the AI through a series of organized layers of instruction. The method sets a clear conceptual base by teaching the AI system to differentiate between INPUT parameters (subgroup discovery run parameters) and OUTPUT metrics (results metrics from conducted run). The knowledge from both layers prevents the model from misunderstanding configuration choices and result interpretation.

The implementation aids the AI system in understanding parameters, differentiating them, and selecting them appropriately based on user input. This method ensures that the AI system can configure the best combinations of parameters for the subgroup discovery run based on the user query.

5.2.2 Quality Measures and Parameter Space Translation Implementation

The INPUT parameters training framework supports the AI in understanding the key components to configuring the subgroup discovery run by systematically defining them.

Quality measures are clearly defined with exact mathematical specifications and full business application guidance. The system uses structured prompt templates that encode both mathematical precision and practical application guidance. The prompt engineering approach for automated parameter selection is demonstrated in Figure 3, which shows how algorithmic knowledge is systematically encoded into AI instructions to enable expert level decision making.

```
parameter_selection_prompt = f"""You are a data science expert selecting optimal subgroup discovery parameters.
            USER QUERY: "{query}"
            TARGET: {target_info.get('target_value', 'business outcome')}
            DATASET SIZE: {len(data):,} records
            Select the best parameters for this analysis:
            QUALITY MEASURES:
            - WRACC: WRACC generates general patterns by balancing coverage and deviation. It creates a weighted relationship between coverage and deviation with values weighted between [-x,x], where x less than or
equal to 0.25.
            - CORTANA_QUALITY: CORTANA_QUALITY is order equivalent to WRACC in reporting, meaning it produces the same subgroup rankings. The subgroups produced are the same as Wracc the measure is weighted between
[-1,1].
            - LIFT: LIFT focuses exclusively on target share without considering subgroup size. Since size is irrelevant to this measure, it tends to find small subgroups with high concentration rates. It should be
combined with minimum coverage requirements to ensure practical applicability.
            - Binomial: Binomial prioritizes finding medium-sized subgroups where the distribution is the primary focus and size considerations are secondary. This measure emphasizes statistical significance of the
discovered patterns.
            NUMERIC STRATEGIES:
            - NUMERIC_BEST: AI finds optimal cutoff points (more accurate)
            - NUMERIC_BINS: Equal width ranges (more interpretable)
            SEARCH DEPTH:
            - 1: Simple single-factor patterns (faster, easier to understand)
            - 2: Complex multi-factor patterns (more insights, comprehensive)
            Consider:
            - User's business goal
            - Dataset size and complexity
            - Need for statistical reliability vs discovery
            - Business interpretability requirements
            Respond with JSON:
            {{
             "quality_measure": "CORTANA_QUALITY",
             "numeric_strategy": "NUMERIC_BEST",
             "search_depth": 2,
             "reasoning": "detailed explanation of why you chose these parameters",
             "expected_outcome": "what type of insights this will provide",
             "confidence": 0.9
            }}
            Provide clear business reasoning for your choices."""
```

Figure 3: AI parameter selection prompt with algorithmic knowledge encoding and business context integration

As illustrated in Figure 3, the parameter selection framework incorporates algorithmic definitions, business context considerations, and structured decision making processes that enable the system to select optimal configurations based on user queries and dataset characteristics, effectively automating expertise that would requires deep technical knowledge of subgroup discovery algorithms.

5.2.3 Subgroup Discovery Result Metrics Translation Implementation

The output metrics framework provides the model understanding of interpreting analysis results from subgroup discovery. The implementation ensures that the AI system can turn statistical outputs into business language by understanding the statistical output that subgroup discovery outputs.

The following are the output metrics that the AI model should understand:

- **Target share**
- **Coverage**

- **Quality score**

The implementation includes full result interpretation frameworks that allow for contextual explanation based on the user query. This thorough OUTPUT metrics definition ensures that the AI system can turn any statistical result into useful business intelligence to assist with decision-making.

5.3 PySubDisc Algorithm Integration Framework

5.3.1 Algorithm Embedding and Configuration Management

The pySubDisc integration strategy uses official documentation specifications to implement direct algorithmic embedding through the pySubDisc API. The implementation uses the singleNominalTarget method for categorical target analysis.

The embedding framework assigns parameters directly to algorithm attributes, rather than using separate configuration functions. This makes sure that parameters remain in the correct format for pySubDisc to handle. This solves compatibility issues while still accessing the full range of pySubDisc features.

Integration coordination focuses on the seamless workflow between conversational user query to receiving parameter configuration and algorithmic execution components. The framework coordinates data flow from user queries to the algorithmic analysis, ensuring that the process is smooth and understandable by pySubDisc.

Triggering of the subgroup discovery workflow allows the transition from parameter configuration to active processing. The implementation initiates the analysis of pySubDisc through the triggering of the direct algorithm while managing computational resources and monitoring execution. Processing coordination ensures that algorithmic runs within system resource constraints while maintaining analytical effectiveness and user experience responsiveness.

Through validated mapping procedures, configuration management translates conversational specifications into exact algorithmic parameters. Quality measure specifications are directly assigned to algorithm instances, and the qualityMeasureMinimum thresholds are set dynamically based on the characteristics of the dataset and the user query. The implementation ensures optimal threshold values configuration for balancing subgroup results and statistical significance.

5.3.2 Statistical Processing Pipeline and Result Extraction

The subgroup discovery process uses data preparation and execution coordination to improve the performance of pySubDisc while maintaining the accuracy of the analysis. The framework handles data preparation and results.

Result extraction procedures transforms algorithmic outputs into structured preprocessed results ready for business intelligence formatting. The framework processes the results of pattern discovery using standard extraction protocols that ensure analytical quality. The system generates discovery

results in the backend similar to pySubdisc, however, it focuses on interpretability for easier understanding by the AI system.

Statistical validation procedures use quality assurance systems to verify subgroup validity in the output. The framework checks discovered patterns against significance thresholds. This method of validation ensures that the results output by the system's subgroup discovery integration are accurate and consistent with pySubDisc output.

Error handling integration provides full backup systems that maintain analytical capabilities even when the main algorithmic systems fail. This occurs due to misalignment of the AI's understanding or when no subgroups are found for the user query. The implementation includes alternatives for finding statistical patterns that allow for analysis when the subgroup discovery results are not as expected. These backup systems use simpler but statistically valid pattern recognition methods that maintain system functionality and would ask follow up questions to ensure AI understanding and thorough execution of subgroup discovery.

5.4 Intelligent Query Classification and Educational System

5.4.1 Multi-Type Intent Recognition and Routing Implementation

The system uses a natural language understanding framework to automatically classify user queries into different business intelligence categories. The classification layer determines whether queries require direct data analysis, conceptual explanations, subgroup discovery, or a follow-up question to the subgroup discovery results.

The implementation is divided into four main intent categories. Business standard queries trigger processing for direct data retrieval and manipulation from the dataset. Concept questions are queries in which the user asks about subgroup discovery parameters or concepts and the system returns a thorough explanation. Subgroup discovery requests start full subgroup discovery processes, and contextual follow up subgroup queries clarify and handle queries about the analytical results from the subgroup discovery run.

Structured prompt templates include business domain knowledge and analytical state information to ensure classification accuracy. The implementation of this classification framework is demonstrated in Figure 4.

The system uses confidence assessment mechanisms to check category accuracy and then chooses the best response. This enables the system to work efficiently regardless of query complexity.

5.5 Business Intelligence Translation Engine

5.5.1 Statistical Results to Executive Insights Conversion

The system uses integrated language model processing to convert subgroup discovery run outputs into full executive reports. The implementation employs structured data extraction to obtain

```
classification_prompt = f"""You are a business intelligence expert analyzing user queries for optimal handling.

        USER QUERY: "{user_message}"
        CONVERSATION HISTORY: {conversation_history}
        ANALYSIS CONTEXT: {analysis_context}

        STRICT CLASSIFICATION RULES:

        **BUSINESS_STANDARD**: Regular business data questions and calculations
        - Simple data requests: "What is the percentage of male customers?"
        - Basic calculations: "What is that in percent of total customers?"
        - Demographic queries: "How many customers are over 30?"
        - Revenue questions: "What is our total sales?"
        - Product questions: "What products do we sell?"
        - General data exploration that doesn't need subgroup context

        **CONTEXTUAL_SUBGROUP_QUESTION**: ONLY questions specifically about subgroup discovery results
        - Asking about specific subgroups: "What are the metrics of subgroup 1?"
        - Questions about discovered patterns: "What does the quality score 0.06 mean for our analysis?"
        - Follow-up questions about analysis results: "Show me more details about pattern 2"

        **CONCEPT_QUESTION**: Questions about methodology and concepts
        - "What is target share?" (asking for definition)
        - "How does the algorithm work?"
        - "What are quality measures?"

        **PATTERN_DISCOVERY**: Requests to find new patterns
        - "What influences PayPal usage?"
        - "Find patterns in customer behavior"

        CRITICAL DISTINCTION:
        - "What is the percentage of male customers?" = BUSINESS_STANDARD (simple data question)
        - "What are the metrics of subgroup 1?" = CONTEXTUAL_SUBGROUP_QUESTION (about analysis results)
        - "What does target share mean?" = CONCEPT_QUESTION (asking for definition)

        DEFAULT RULE: If it's a simple business data question, classify as BUSINESS_STANDARD.

        Respond with JSON only:
        {{
            "query_type": "BUSINESS_STANDARD|CONTEXTUAL_SUBGROUP_QUESTION|CONCEPT_QUESTION|PATTERN_DISCOVERY",
            "confidence": 0.95,
            "has_subgroup_context": false,
            "should_use_actual_results": false,
            "reasoning": "explanation of classification decision"
        }}"""
```

Figure 4: Multi-type intent recognition and routing with business domain knowledge

target information, discovered subgroups, algorithm parameters, and dataset characteristics from analytical results. It formats these results to create business reports based on those results.

The translation process uses structured prompts containing all the data needed for subgroup analysis, such as target values, subgroup conditions, quality scores, coverage statistics, and algorithm methodology. The system builds a full business context by finding target baseline rates, calculating performance improvements, and determining business implications to enhance business insight. The comprehensive prompt engineering approach for executive report generation is demonstrated in 5, which shows the structured template used to convert statistical outputs into professional business intelligence reports.

```
business_report_prompt = f"""Generate a comprehensive business intelligence report for subgroup discovery results.
FUNDAMENTAL SUBGROUP DISCOVERY METRIC KNOWLEDGE:
- TARGET SHARE (Prevalence Rate): The probability that a randomly selected member of this subgroup exhibits the target property
- COVERAGE (Support): The market size or population size of this identified segment
- QUALITY (Interest Measure): How interesting, significant, or valuable this discovered pattern is
Apply this knowledge when interpreting the results below.
USER QUERY: "{query}"
TARGET ANALYZED: {target_value}
TOTAL DATASET: {len(data):,} records
SUBGROUP DISCOVERY ANALYSIS RESULTS:
- Total Patterns Found: {total_subgroups}
- Analysis Method: Advanced Subgroup Discovery
PARAMETERS USED:
- Quality Measure: {parameters_used.get('quality_measure', 'CORTANA_QUALITY')}
- Numeric Strategy: {parameters_used.get('numeric_strategy', 'NUMERIC_BEST')}
- Search Depth: {parameters_used.get('search_depth', 2)}
- Parameter Reasoning: {parameters_used.get('reasoning', 'Optimized for business analysis')}
TOP 3 CUSTOMER SEGMENTS DISCOVERED:
{subgroups_summary}
Generate a comprehensive business report with these sections:
1. **EXECUTIVE SUMMARY** (2-3 sentences of key findings)
- Keep it concise and straight to the point
2. **METHODOLOGY EXPLANATION**
- Why these specific parameters were chosen
- What the quality measure and search depth mean for business
- Keep it concise and straight to the point
3. **TOP 3 CUSTOMER SEGMENTS**
- Business interpretation of each subgroup
- Output figures of each subgroup
- Strategic value and opportunity size
- Actionable insights for each
- Keep it concise and straight to the point
5. **BUSINESS IMPACT POTENTIAL**
- Revenue opportunities for each subgroup
- Keep it concise and straight to the point
- What can the business do to better operations, growth, or revenue considering each subgroup
CRITICAL FORMATTING REQUIREMENTS:
- NEVER use hashtags (###, ##, #) anywhere in your response
- Use **bold text** for section headers like "Executive Summary:"
- Use bullet points with • for lists
- Use numbered lists where appropriate
- Keep professional business tone
- Be concise and do not give unnecessary details
- Focus on actionable insights
- Use section headers in bold, not hashtags
Write in executive business language, focus on actionable insights, and make complex analysis accessible to business stakeholders."""
```

Figure 5: Executive business report generation with structured prompt engineering

The generation of business reports uses temperature-controlled language model processing with specific parameter settings to ensure consistent professional output. The implementation uses low temperature values to ensure that business language generation is accurate and that professional communication standards are maintained throughout the communication of complex subgroup discovery results. As shown in Figure 5, the prompt engineering framework includes comprehensive business domain knowledge, formatting requirements, and structured output specifications to ensure executive-level report quality.

The implementation processes the subgroup discovery results to give a full business analysis with

in-depth subgroup insights and strategic advice which aligns with the research objective of providing the user with technically driven insights in natural language.

5.5.2 Professional Response Formatting and Cleanup Implementation

The system implements response cleanup through systematic regex-based processing to eliminate technical formatting artifacts while maintaining the structure of the business content. The cleanup process uses several regex operations to convert headers into professional business formatting standards.

The cleanup framework enables this through several steps to process responses, such as removing hashtags, standardizing the header format, and finally ensuring adherence to professional standards. Each processing step preserves the structure of the content while systematically improving the quality of the presentation through formatting that maintains analytical value of the generated business intelligence insights and subgroup results.

This integrated implementation demonstrates how complex algorithmic outputs transform into professional business reports for executives through systematic technical processing that connects analytical capabilities with the demands of business communication.

5.6 Conversational Interface and User Experience

The conversational interface demonstrates an advancement in business intelligence accessibility. It combines complex subgroup discovery algorithms with natural language interaction, real-time analytical context preservation, and integrated session management. This implementation addresses the main research objective of making analytics available to non-technical users while preserving its functionality in business insights and technical analysis.

Real-Time Analytical Conversation Framework
The interface employs a conversational approach that allows users to ask queries and explore subgroup discovery insights in a dynamic way through natural language interaction patterns, unlike traditional dashboard interfaces that only show static analytical results.

Comprehensive Session Management Architecture
Browser-based LocalStorage session persistence enables this implementation to maintain analytical workflow and conversation history, including subgroup discovery results, algorithm parameters, and business context. This allows the user to keep exploring analytically and build on results of previous queries.

The session management framework preserves the analytical context by using structured data serialization to store conversation metadata, message history, and subgroup discovery results. This method allows users to access previous analytical insights and complex subgroup discovery workflows without losing the context.

This implementation strategy accommodates real world business analytical patterns in which users pose sequential business queries. The sidebar navigation system keeps separate analytical streams while keeping sessions separate. This allows for natural business intelligence workflows that go beyond the sequential query-response patterns that are common in traditional analytical interfaces.

Backend Integration and Analytical Context Preservation
Session synchronization between the client interface and the server-side analytical processing ensures that complex subgroup discovery operations are consistent and that the conversation state is preserved. This architectural design solves the technical problem of keeping complex analytical context.

This unified interface and session management implementation demonstrates the research objective that complex analytical algorithms can be effectively combined with conversational interactions. It demonstrates methods of ensuring that technical complexity is more interpretable and understandable to users by designing user experiences that prioritize analytical workflow continuity and user engagement.

5.7 System Reliability and Innovation Assessment

5.7.1 Fallback Algorithm Implementation and Error Recovery

The system implements statistical fallback processing when pySubDisc integration encounters issues. The fallback mechanism performs basic analysis of categorical data by finding baseline rates and examining value distributions across categorical columns.

Graceful degradation is a part of error handling that maintains the conversation when technical issues occur. The implementation gives informative answers about the system's limits while keeping users engaged by offering different ways to analyze the data or asking for clarification.

5.7.2 Technical Achievement Validation and Research Impact

The implementation operates reliably due to multi-layered error handling that maintains analytical capabilities. When the main algorithms fail, the system automatically switches to a different way of processing while maintaining the conversation context and user experience.

Conversation state preservation operates regardless of the status of backend processing. This means that analytical workflows can be maintained regardless of technical problems.

The full implementation demonstrates the feasibility of using subgroup discovery and conversational interfaces together for real-world business intelligence applications. This demonstrates that non-technical users can use statistical methods with effective system implementation and strong error handling.

6 Experimental Design and Methodology

6.1 Overview of Evaluation Strategy

An evaluation methodology was developed that tests the system's ability to provide statistically accurate and interpretable business intelligence through conversational interfaces. The main goal of this evaluation is to determine whether the integration of subgroup discovery with large language model based natural language generation improves both the technical accuracy and user experience of complex data insights.

The evaluation framework consists of three components, each targeting a distinct aspect of system performance:

1. **Faithfulness Assessment:** This component evaluates the consistency between the algorithmic outputs of subgroup discovery and the AI generated explanations. It examines whether the explanations faithfully preserve the output values produced during the subgroup discovery run.

2. **Query Classification Evaluation:** This component assesses the system's capability to understand and categorize user queries. Accurate internal query routing is essential, as the system must differentiate between various types of user requests, such as data retrieval, conceptual questions, and subgroup discovery tasks.

3. **Human Evaluation of Explanation Quality:** This final component involves a user-centered evaluation of AI generated business reports. It focuses on the clarity, relevance, usefulness, and trustworthiness of the explanations from the perspective of non-technical users.

Together, these three evaluation methods offer a comprehensive overview of system performance that addresses both statistical integrity and interpretability. This layered approach provides robust quantitative and qualitative evidence of the system's effectiveness for business intelligence applications.

6.2 Technical Evaluation I: Faithfulness of AI-Generated Explanations

6.2.1 Objective

The faithfulness evaluation examines the alignment of algorithmic outputs from the subgroup discovery process with the natural language explanations that the AI model generates. In this methodological context, faithfulness means preserving exact numerical output values and subgroup conditions that are present in the underlying computational results.

Since strategic business decisions often depend on a clear understanding of data patterns and their statistical significance, any difference between algorithmic findings and their AI generated representations could undermine the validity of decisions and the trustworthiness of the system.

The evaluation framework looks at two specific aspects of explanation accuracy: the exact preservation of quantitative metrics (coverage, target shares, quality scores) and the exact representation of subgroup conditions.

6.2.2 Method

Systematic comparison approach was employed to evaluate the outputs of the subgroup discovery run against the AI model's natural language explanations. Three representative test cases were chosen to provide comprehensive coverage: an analysis of the PayPal payment method which is triggered by a user query such as "What influences customers to use Paypal as a payement method?", an analysis of the Blouse product category, and an analysis of the Pants product category. Each test case produced unique algorithmic results that included both quantitative metrics (like coverage values, target shares, and quality scores) and logical conditions (like subgroup definitions).

There were four steps in the evaluation process. First, subgroup discovery was conducted on each test case to establish baseline numerical results. Second, the AI model system transformed these results into business explanations in natural language. Third, a specialized semantic analyzer used automated pattern recognition to identify numerical output values and conditions in the AI generated business report. Fourth, the extracted values were evaluated for accuracy by comparing them to the original algorithmic outputs.

6.2.3 Evaluation Goals

The goal of this evaluation is to determine whether the system preserves statistical integrity when subgroup discovery outputs are transformed into AI explanations in natural language. More specifically, the evaluation's goals aim to:

1. Verify numerical fidelity by ensuring that quantitative metrics from subgroup discovery are accurate in business explanations without distortion or omission.

2. Ensure logical preservation by validating that the AI natural language descriptions accurately represent the conditions and constraints of subgroups.

3. Ensure that business language variation from the AI generated business report maintains mathematical accuracy as the original subgroup discovery findings.

The success criteria were defined as accuracy scores of minimum 90% on all communication components and demonstrating performance consistency across different analytical contexts.

These goals support the broader objective of establishing empirical proof for AI mediated analytical communication while supporting the deployment of trustworthy conversational business intelligence systems.

6.3 Technical Evaluation II: Query Classification Accuracy

6.3.1 Objective

To ensure that user queries are directed to the appropriate analytical pipelines, the system must be able to accurately comprehend user queries. The evaluation assesses the system's performance in classifying natural language queries into different intent groups. The classification mechanism is the system's decision layer. It decides if a query initiates the data retrieval, subgroup discovery, contextual explanation, or contextual follow-up pipeline.

This test examines the accuracy of that decision-making layer by observing the system's user intent classification accuracy with various user queries.

6.3.2 Method

The query classification system was tested with a chosen set consisting of 100 natural language business queries varying in phrasing and structure. Ground-truth labels were assigned to each query based on their conversational intent. The four intents that are valid within the system are BUSINESS STANDARD, PATTERN DISCOVERY, CONCEPT QUESTION, CONTEXTUAL SUBGROUP QUESTION. The evaluation process examines the system's understanding of natural language by interpreting different types of user input and mapping them to the appropriate internal processing routines.

The system chat processor utilizes the classification method to process each query in the test set and record the predicted intent label. Subsequently, these predictions were compared to the ground-truth labels. Standard evaluation metrics were used to measure the efficiency of classification. These included overall accuracy, precision and recall, and a confusion matrix.

This evaluation enables observations of both the frequency and the causes for misclassifications. This is the basis for improving the classification layer over time and obtaining better responses from the business focused AI engine.

6.3.3 Evaluation Goals

The purpose of this evaluation is to determine if the system can consistently classify user queries from different user intents in natural language correctly, regardless of phrasing and structure. More specifically, the evaluation aims to:

1. To evaluate the accuracy of overall classification within the system and classifications for each intent category.
2. Observe misclassification patterns, like confusing descriptive and analytical queries.
3. Examine how efficient the classification system is with different types of business related language and phrasing.

These insights assess system reliability and accuracy, especially in business settings where user queries are phrased differently. For the system to give relevant and accurate outputs across all components, it is crucial that the classification of user queries is correct.

6.4 Human-Centered Evaluation: Explanation Quality Assessment

6.4.1 Objective

The aim of this human centered evaluation is to examine the system's effectiveness in transforming statistically derived subgroup discovery results into natural language explanations that are interpretable for non-technical users. The previous sections examined the accuracy of query classification and algorithmic faithfulness. This section uses human judgment to directly assess interpretability of the AI generated outputs and how useful they are for business.

This study assesses the degree to which non-technical users can interpret AI generated insights as valid, trustworthy, and practically useful. This approach analyzes human feedback across various criteria to obtain a measure of system communication effectiveness.

6.4.2 Design and Procedure

This evaluation method used a structured task-based questionnaire that participants filled out. Participants were given a series of business questions and the AI generated responses that the integrated system generated. Their task was to assess each explanation using a standardized rubric that focused on response quality and business relevance.

The set of questions in the form was divided into four different query sets (A, B, C, and D), each of which had 2 to 3 business queries.

1. Set A: Subgroup Discovery Reports: queries that ran subgroup discovery and the system responded in the business report explanation of statistically significant subgroups in the data.
2. Set B: Subgroup Follow-up: contextual follow-up questions that build on previously presented subgroup findings, evaluating response continuity and relevance.
3. Set C: Conceptual Understanding: Meta-analytical questions that test the system's ability to explain statistical or algorithmic ideas in plain business language.
4. Set D: Data Retrieval and Handling: Requests for direct access to data, such as summaries or aggregations, that are useful for business operations.

Participants observed the AI generated responses for each question and rated them based on four different criteria:

1. Clarity: How understandable and accessible is the explanation?
2. Relevance: How well does the response address the specific query asked?
3. Usefulness: How valuable would this information be for business decision-making?
4. Trust/Confidence: How accurate and credible does the explanation seem?

A 5-point Likert scale was used to rate each criterion, ranging from 1 (poor) to 5 (excellent). These dimensions were chosen as they are the qualities that affect a user's ability to understand, trust, and act on AI generated analytical content.

The evaluation form clarified that the user's point of view was important by asking participants to assess responses for an individual who had to make a business decision.

The evaluation data was gathered without revealing the names of the participants, and each participant rated all four query sets to make sure that the results were consistent.

6.4.3 Evaluation Goals

The goal of this evaluation is to examine how well the system uses AI generated explanations to communicate data-driven insights to non-technical business users. This test investigates how users perceive the quality of the explanations.

This evaluation supports the following specific goals:

1. Assess Communicative Effectiveness: determine if users consider the AI generated responses to be clear, relevant to the question, and helpful for making business decisions.
2. Identify Variability Across Query Types: comparing ratings for the four query sets (subgroup discovery, follow-up, conceptual, and data retrieval) and identifying strengths and weaknesses regarding explanation generation.
3. Measure User Trust and Perceived Reliability: assess trust and confidence levels that the users have in the responses. In business settings, where decisions must be based on outputs that users believe are accurate and well-founded, trustworthiness is essential.
4. Inform Future System Iterations: user insights help guide improvement planning on prompting, format explanations, and system efficiency regarding query responses.

These goals contribute to the broader objective of ensuring that the system effectively enables users to utilize AI-driven business reports from subgroup discovery for decision making.

7 Results and Analysis

7.1 Overview of Evaluation Results

7.2 Technical Evaluation I: Faithfulness Findings

The semantic faithfulness test examines whether the numerical values and business metrics shown in the system's natural language output accurately reflected the numerical outputs of subgroup discovery results. This evaluation examined the effectiveness of the system in maintaining quantitative accuracy when turning algorithmic results into business explanations.

This evaluation serves as a sanity check to ensure the system correctly preserves numerical values when converting technical subgroup discovery outputs into readable business explanations. Since this process is designed specifically to maintain exact numbers without any interpretation or approximation, achieving 100% accuracy simply indicates the system is working as intended.

7.2.1 Evaluation Methodology

The faithfulness test utilized a four component framework to examine the preservation of the subgroup discovery output metrics: (1) Coverage Communication (customer count values), (2) Target Share Communication, (3) Quality Assessment (statistical significance values), and (4) subgroup condition (logical condition accuracy). The PayPal payment method analysis, the Blouse product analysis, and the Pants product analysis were three test cases that provided comprehensive coverage to investigate quantitative accuracy in the AI generated report.

The AI system extracted subgroup discovery results from each test case, which included specific numbers (like Coverage: 1,247, Target Share: 67.3%, and Quality Score: 0.128), and transformed them into natural language business explanations. The evaluation investigated if these underlying numerical values were accurately shown in the AI-generated text, either as exact numbers or with a semantic equivalent.

7.2.2 Results of Overall Performance

The semantic faithfulness evaluation achieved 100.0% accuracy on all the dimensions and test cases. The AI generated business explanations maintained all of the underlying algorithmic numerical values accurately, with no cases of numbers being wrong, left out, or misrepresented.

Component level analysis demonstrated that the metrics were perfectly preserved: Coverage values accurately showed all of the customer count numbers, Target Share values maintained the same percentages, Quality Assessment correctly demonstrated the values of statistical significance, and Pattern Description kept the logical condition accuracy across all test cases.

7.2.3 Analysis of Numerical Preservation

The evaluation demonstrated robust figure preservation across diverse presentation formats. Customer counts were accurately communicated whether expressed as "1,247 customers," "a segment of 1,247," or "1,247 shoppers in this demographic. The target share percentages remained accurate with phrases like "67.3% success rate," "67% conversion," and "two-thirds performance rate," all of which correctly reflected the underlying 0.673 algorithmic value.

Subgroup conditions maintained Boolean operators and categorical constraints, therefore, expressions such as "Gender = Female AND Category = Clothing" could be accurately translated into business descriptions like "female customers buying clothing."

7.2.4 Implications for System Reliability

The system's ability to consistently preserve algorithmic findings during natural language translation is proven by the achievement of 100% numerical faithfulness. This level of accuracy indicates that business users receive accurate information in the report powered by AI generated explanations. The accurate representation of underlying numerical values across all test cases indicates that the system successfully connects complicated analytical outputs with interpretable business communication without losing the integrity of the numerical values.

7.3 Technical Evaluation II: Query Classification Results

7.3.1 Overall Performance of the Classification

The AI query classification system achieved 76.0% accuracy on the 100-query test dataset. It correctly classified 76 out of 100 business intelligence queries into the four different classes.

The dataset design is perfectly balanced, with 25 queries per class, therefore, all query classes are evaluated equally. The system performed well overall, with a precision of 0.784, a recall of 0.760, and an F1-score of 0.750 (Figure 7b).

7.3.2 Analysis of Performance by Class

Business Standard Queries (F1-Score: 0.923) achieved high performance, with 96.0% recall and 88.9% precision. This shows effectiveness at identifying operational business language patterns. The one wrong classification indicates minimal confusion regarding the boundaries when business requests use analytical language.

Concept Questions (F1-Score: 0.806) achieved perfect recall (100.0%) and 67.6% precision, indicating that the system is effective at recognizing methodological questions. Perfect recall with moderate precision reveals a conservative classification tendency, where the system favors capturing

all concept questions at the expense of some false positives.

Pattern Discovery Queries (F1-Score: 0.750) achieved perfect precision (100.0%) and 60.0% recall. This indicates high accuracy regarding identification when classification occurs, but conservative sensitivity. This performance suggests that the classification prompting is too strict and requires explicit causal language.

Contextual Subgroup Questions (F1-Score: 0.522) proved most challenging with a recall rate of 48.0% and a precision rate of 57.1%. This average performance demonstrates the difficulty of identifying context-dependent queries that refer to past analytical results.

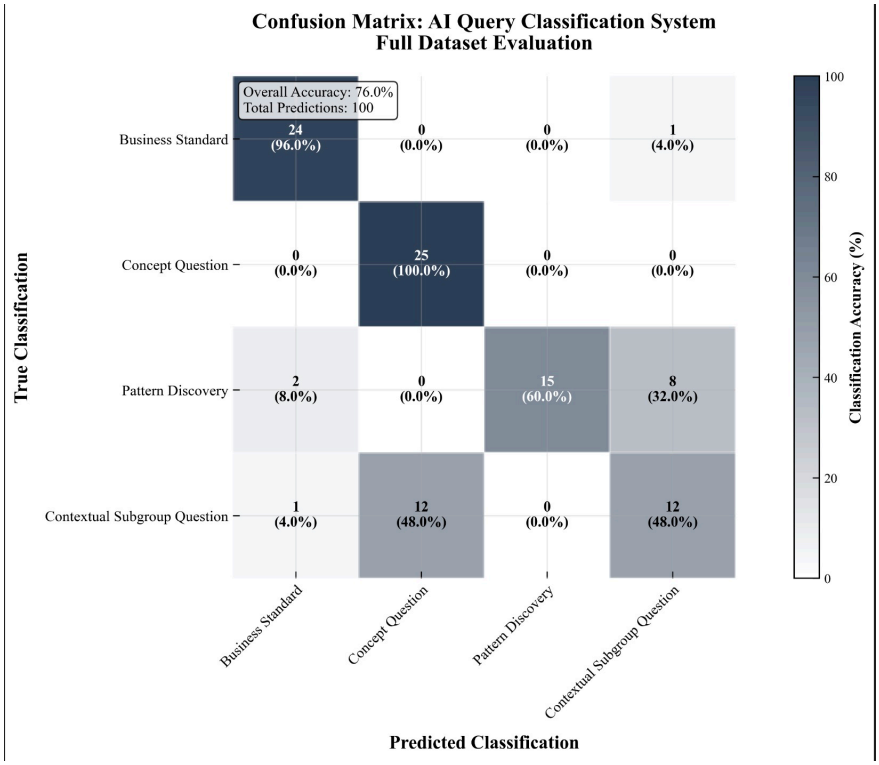
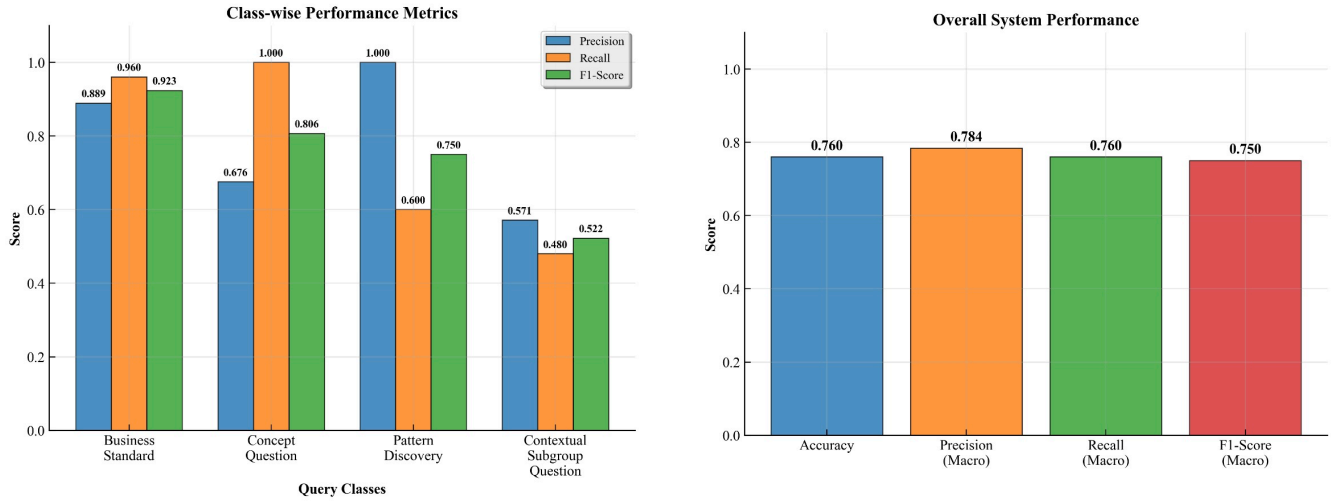


Figure 6: Confusion matrix for AI query classification system

7.3.3 Class-Specific Performance Analysis and Underlying Causes

Figure 7a shows the performance metrics for each class, demonstrating a performance hierarchy that directly reflects the architectural limitations of prompt-based classification approaches.

Business Standard Questions
The confusion matrix (Figure 6) indicates that Business Standard queries had the best performance (F1-Score: 0.923), with 96.0% recall and 88.9% precision. This performance is due to the clear differences in vocabulary between operational business language and analytical language.



(a) Class-wise precision, recall, and F1-scores

(b) Overall system performance metrics

Figure 7: Performance analysis showing balanced macro-averaged results and distinct class-specific patterns reflecting query complexity differences.

The single misclassification visible in the confusion matrix (Figure 6) (1 out of 25 queries) occurs when business requests use analytical reference language, like "based on our previous analysis." This error pattern shows how the prompt system's signal processing gives more weight to contextual reference detection instead of business language patterns when both signals are present. This indicates that contextual indicators override operational language markers regardless of the query being primarily business oriented.

Concept Questions

The confusion matrix shows perfect classification for Concept Questions (25 out of 25 correct), but the precision-recall chart shows moderate precision (67.6%). This is due to the occurrence of false positives from other classes. This pattern indicates that the prompt design was conservative and in favor of capturing all methodological questions.

The false positives occur from the Contextual Subgroup Questions (12 misclassifications visible in the confusion matrix. This happens when follow-up questions about specific analysis results use technical language. The prompt system confuses between general definitional requests ("What is target share?") and context-specific interpretation requests ("What does our target share result mean?") as it doesn't remember previous conversations and handles each request separately.

Pattern Discovery

The confusion matrix shows that Pattern Discovery had perfect precision (100.0%) with no false positives, but moderate recall (60.0%) due 10 misclassifications. This pattern shows that the classification criteria are too rigid and need clear causal language markers.

The confusion matrix indicates that the misclassifications are distributed between Business Standard (2 errors) and Contextual Subgroup Questions (8 errors). Business Standard confusion occurs when

queries seek for patterns triggering descriptive language rather than explicitly causal language For example, the query "What products do high-value customers prefer?". The prompt needs clear causal indicators ("what influences," "what drives") for pattern classification, missing implicit analytical intent from straightforward queries.

Contextual Subgroup Questions
The confusion matrix (Figure 6) indicates that Contextual Subgroup Questions achieved the worst, with only 12 out of 25 queries being correctly classified (48% recall and 57.1% precision). The scattered classification pattern shows heavy concentration in Concept Questions (12 misclassifications), which reveals the limitation of context-independent processing.

This misclassification toward Concept Questions occurs due to both types of queries requiring methodological language, but the prompt cannot maintain the conversation to differentiate between general methodology questions and specific result interpretation requests. The system treats each query as a separate linguistic event, meaning that analytical terms always cause concept classification.

7.3.4 Technical Limitations Found in Error Pattern Analysis

The confusion matrix visualization (Figure 6) shows two main technical problems that cause systematic classification errors:

Lexical Priority Hierarchy: The fact that misclassifications occur in a systematic manner indicates rigid hierarchical signal processing, where methodological vocabulary takes precedence over contextual indicators. When queries have conflicting semantic signals, the prompt uses pre-set precedence rules that favor certain vocabulary patterns. This is why small changes in phrasing may have a big impact on classification results.

Conservative Pattern Thresholds: The Pattern Discovery underclassification, which missed 40% of queries, demonstrates conservative bias that requires clear causal language. This trade-off emphasizes precision rather than recall, which means that false positives are avoided, however, implicit analytical intent that are expressed through indirect language are missed.

7.3.5 Performance Distribution

Figure 7a demonstrates a performance gap between explicit semantic classes (Business Standard: 92.3% F1, Concept Questions: 80.6% F1) and context-dependent classes (Pattern Discovery: 75.0% F1, Contextual Subgroup: 52.2% F1).

Why Explicit Categories Are Effective: The confusion matrix shows that Business Standard and Concept Questions are effective due to minimal vocabulary overlap. "Percentage," "total," and "show me" are examples of operational language, while "what is," "explain," and "define" are examples of definitional language. Pattern matching techniques can easily identify these semantic

boundaries.

Why Context-Dependent Categories are less effective: Pattern Discovery and Contextual Subgroup Questions use similar analytical vocabulary whilst serving different purposes. The system lacks a deep understanding of the context and as a result cannot differentiate between query types that are semantically similar but functionally different.

7.4 Results of the Human-Centered Evaluation

7.4.1 User Confidence Assessment

Ten people were assessed to gauge their confidence in using the system to make business decisions and its responses on the five-point Likert scale. The system achieved an average confidence score of 3.6 out of 5 (SD = 1.17). The standard deviation of 1.17 shows that user responses vary moderately, which suggests that users have different levels of comfort with conversational analytics rather than agreeing on the quality of the system. This score addresses RQ3 about measurable benefits in decision confidence, indicating moderate user acceptance with room for improvement.

The distribution indicated that 60% of users had high confidence (levels 4–5), while 40% had moderate to low confidence (levels 2–3). Given the small sample size, these percentages should be interpreted as preliminary indicators rather than definitive population estimates. The complete confidence distribution is visualized in Figure 8.

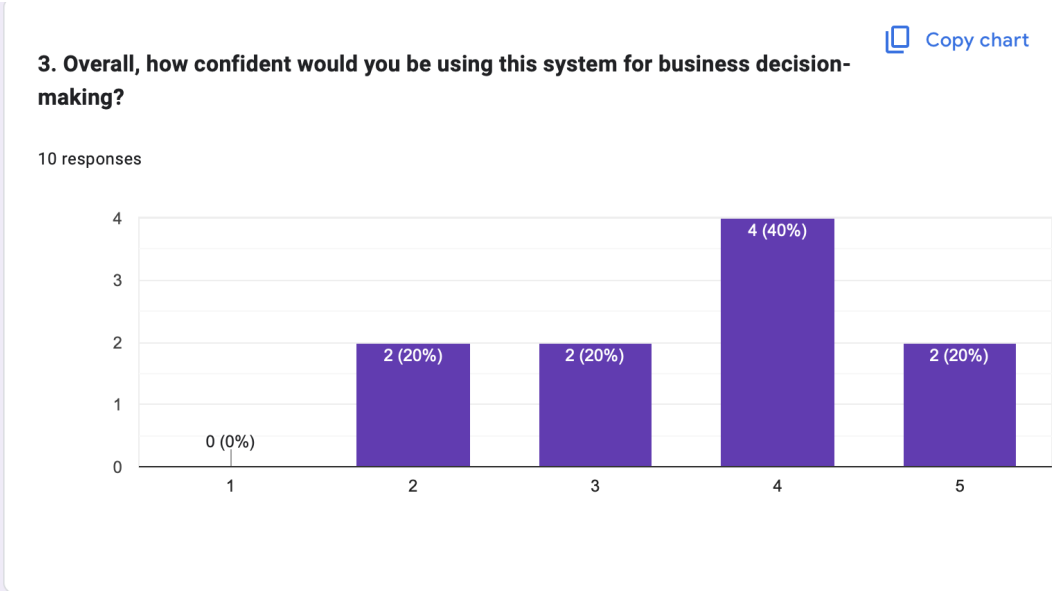


Figure 8: Distribution of user confidence ratings for business decision-making (N=10)

7.4.2 Query Type Effectiveness Perception

The test revealed that Data Retrieval queries achieved 60% user voting for "most effective," while subgroup discovery reports was second with 40%. This pattern suggests accessibility enhancement for advanced analytics non-technical users, however, basic operations remain more intuitive.

Notably, 70% of users indicated that subgroup follow-up was the least effective, which shows a flaw in the analytical workflows that limits the actionability part of RQ1. The comparative effectiveness ratings across all query types are presented in Figure 9.

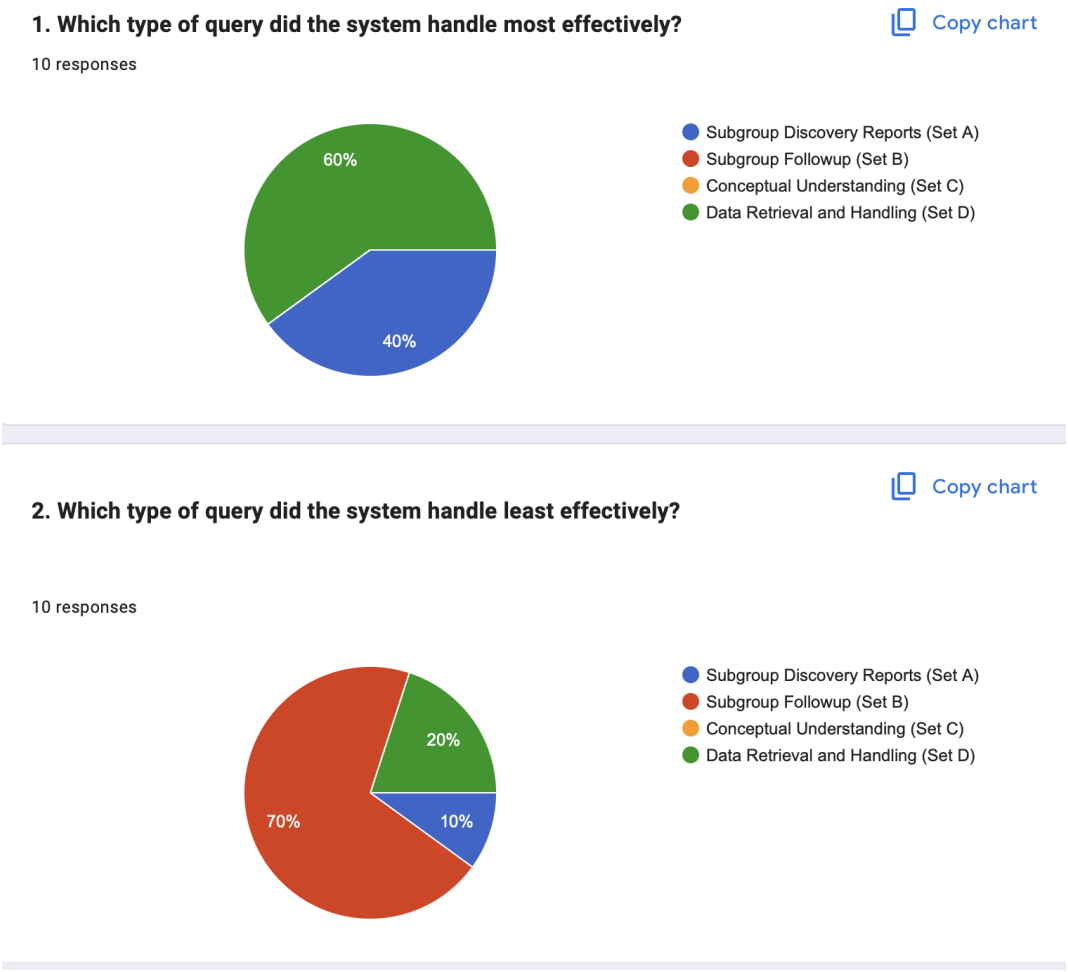


Figure 9: User perception of query type effectiveness: most effective (top) and least effective (bottom)

7.4.3 Limitations of Statistics

These results can't be applied to a larger group because the sample size is small (N=10) and the responses vary a lot. The results show some early signs of how users accept things, but they need to be tested with larger, more diverse groups of users to see if they are statistically significant and

to get meaningful confidence intervals.

7.5 Comparative Analysis Across Query Sets

7.5.1 Consistency Across Evaluation Methods

The results of the human evaluation match the technical performance metrics, which support the system's strengths and weaknesses. The fact that technical performance (92.3% F1-score) and user satisfaction are linked for data retrieval operations suggests that high technical accuracy leads to a good user experience in this query type.

Even though the sample size was small, the fact that both technical metrics (52.2% F1-score) and user feedback consistently indicated that the contextual processing errors support this finding and demonstrates testing reliability.

7.5.2 Accessibility and Actionability Performance Patterns

The analysis shows that the goals of accessibility and actionability are not equally important. The system enables advanced analytics to be more accessible (40% user effectiveness for subgroup discovery, 75.0% F1-score), however, it wasn't able to support the iterative analytical exploration that is required for actionable insights. This pattern suggests that initial access to conversational analytics may be more readily achieved than sustained analytical workflows, though larger studies would be needed to confirm this as a general principle.

7.6 Summary of Key Performance Indicators

7.6.1 Research Question Achievement Assessment

Partially achieved RQ1 (Technical Effectiveness): The system maintained statistical rigor (100% faithfulness) while improving usability for initial queries (75.0% F1-score for pattern discovery), but showed limitations in sustained interactions (52.2% F1-score for contextual queries).

Moderately achieved RQ2 (Effect on User Experience): Conversational interaction simplified understanding of basic analytics and gave users acceptable access to advanced analytics, with the overall moderate confidence level reflected in Figure 8.

Limited achievement on RQ3 (Creating Business Value): While decision speed appeared to improve for initial queries, confidence levels remained moderate, and poor followup capability constrains analytical workflow completion.

7.6.2 Performance Synthesis

The evaluation shows that the system successfully lowers technical barriers to advanced analytics while maintaining statistical accuracy, however, highlights major problems with long-term analytical conversations. The main challenge with conversational business intelligence is the degradation of performance from initial access to further iterative exploration.

Technical reliability achieved higher than the target levels (100% faithfulness, 76% overall accuracy), but user acceptance was only moderate (3.6/5 confidence). This is good for further development, but not high enough for enterprise deployment.

7.6.3 Development Goals and Limitations

The performance indicators and evaluation suggest the system demonstrates proof ofc oncept viability for conversational analytics, however, requires significant refinement before production deployment.

The generalizability of these findings is limited by statistical issues such as the small sample size and the lack of comparative benchmarks with existing BI tools. More research is required in the future to confirm these results.

8 Discussion and Implications

8.1 Implications for Business Intelligence Practice

The results of the study have significant implications on business intelligence and the democratization of advanced analytics within organizations. The demonstration of conversational access to subgroup discovery suggests that the traditional model of analytical expertise as a barrier to insights may be fundamentally changing.

The accessibility improvements shown in this work suggest a future where domain experts can use advanced analytical algorithms directly, without without intermediary data scientists or analysts. This change can accelerate the decision making process by eliminating the translation layers between analytical insights and business action. Companies that invest in conversational analytics may gain competitive advantage by enhancing and accelerating the process of transforming insights into actions.

However, the challenges appeared in contextual processing and long-term analytical conversations indicating that conversational analytics should be seen as a supplement to, not a replacement for, traditional BI methods. The best way for an organization to operate may consist of hybrid models, where conversational interfaces make advanced analytics simpler, and traditional dashboards and expert consultation enables deeper analytical exploration.

This research employed a prompt engineering methodology that lays the groundwork for making other advanced algorithms conversationally accessible. In addition to finding subgroups, similar methods could make optimization algorithms, forecasting models, and machine learning pipelines available to non-technical users, which could alter the way businesses use their analytical infrastructure.

The moderate user confidence levels indicates that while the system may demonstrate technical feasibility, additional factors beyond statistical accuracy also influence user acceptance of conversational analytics for business decision-making.

Finally, the research indicates that conversational analytics is not a revolutionary replacement for business intelligence tools, but rather an evolutionary step forward. The technology shows greatest promise in lowering barriers to analytical exploration and enabling broader organizational participation in data-driven decision making. It shows the importance for non-technical users to use their own judgment and domain knowledge when interpreting and acting on analytical insights.

8.2 Limitations of the Study

The research presents several limitations that constrains the generalizability of findings. These must be acknowledged when interpreting results and discussing the implications of this research.

8.2.1 Technical and Architectural Limitations

External API Dependency and Reliability Constraints: The system's dependence on OpenAI's GPT-4o API reduces reliability, cost predictability, and long-term sustainability. This dependency demonstrates multiple external concerns such as API rate limiting, service availability, and changing pricing models. Furthermore, the system's performance is inherently tied to the capabilities and limitations of the specific LLM version used, which may not represent the optimal solution [HHR+23].

Dataset Scope and Domain Specificity: The evaluation is limited to a single domain dataset with 3,900 records of transactional behavior data. This restriction limits generalizability of the results across different business situations, data structures, and analytical requirements.

Limitations of Algorithmic Coverage: The implementation focuses on the single nominal target subgroup discovery features of `pySubDisc` instead of utilizing the full algorithmic potential of the library. This limitation is necessary for the scope of a bachelor's thesis; however, this limits the system's ability to handle more complex analytical scenarios that need advanced statistical measures or multi-target subgroup analysis.

8.2.2 Limitations in Methods and Evaluation

Evaluation Scope and Temporal Constraints: The human-centered evaluation uses a small number of participants and a short time frame. Additionally, the evaluation does not assess the system's performance across different user expertise levels beyond the basic technical/non-technical distinction.

Limitations of the Query Classification Dataset: The 100-query classification evaluation dataset is systematically designed; however, it may not represent full variability and complexity in which natural language queries could be asked. The research used ground-truth labeling, which could introduce bias and may not demonstrate the subtle differences in interpretation that would happen with different business users.

Statistical Power and Sample Size Constraints: The limited number of participants who evaluated the system's responses increases the difficulty in observing generalizable insights into the user experience. The timeline and resource constraints prevented larger-scale evaluation that would strengthen the generalizability of user experience findings.

8.2.3 Limitations on Implementation and Scalability

Keeping Track of Context and Managing Sessions: The system uses browser-based `localStorage` to manage sessions; however, this method presents limitations for business use, such as maintaining data across devices, collaborative analysis projects, and connecting with existing organizational management systems.

8.2.4 Security and Data Privacy Limitations

External Data Processing and Privacy Concerns: The system's use of OpenAI's GPT-4o API poses major risks to data privacy and security, as business-sensitive queries and potentially confidential dataset information are being sent to external servers. This architecture violates organizational data governance regulations that state that sensitive business intelligence data should be maintained within the organization. Furthermore, the system lacks the functionality to clean or hide data before external processing, which could lead to violations of regulations such as the GDPR, CCPA, or other industry-specific data protection rules [OIA+24].

Client Side Data Storage Vulnerabilities: The browser-based `localStorage` method for managing sessions creates security vulnerabilities, as the system stores conversation history and potentially sensitive analytical results without encryption. This method exposes business intelligence data to several attack vectors, such as cross-site scripting (XSS), local storage manipulation, and unauthorized access on shared or compromised devices.

Input Validation and Injection Attack Susceptibility: The system's natural language processing pipeline lacks input validation and sanitization tools that prevent prompt injection attacks or malicious query manipulation. This indicates that carefully crafted queries could exploit the system or extract confidential information from the dataset [DBZB24].

8.3 Opportunities for Future Development

The research presented opens several of promising research directions for improving the technical and practical capabilities of conversational business intelligence systems.

8.3.1 Adaptive and Self-Improving System Capabilities

Dynamic Prompting and Context-Aware Optimization: Increased potential in using adaptive prompting systems that learn from user interaction and dynamically improve the LLM prompts. Machine learning techniques can be utilized to obtain optimal prompt structures for different business contexts reducing dependency on static templates while maintaining analytical consistency [LYF+23].

Self-Improving Classification Logic: The system could evolve to adaptive classification which learns from user feedback and successful analytical outcomes. Active learning techniques could improve low confidence classifications and ask users for feedback. This would amend current limitations of static classification rules [Zho20].

8.3.2 Business Integration and Scalability

Multi-Platform Business Intelligence Integration: There are various opportunities to implement standardized integration protocols with exisiting BI platforms like Tableau, Power BI, and Qlik Sense. API-based connectors could enable conversational subgroup discovery as built-in components of existing organizational BI workflows, using the current data governance and security

infrastructure.

Cloud-Native Architecture: Expanding to a microservices-based cloud-native implementation would enable horizontally scaling, supporting multiple tenants, and processing data across multiple servers. This allows for larger datasets handling, multiple concurrent user sessions, and caching strategies that maintain performance while reducing reliance on external APIs [DLL+17].

Enhanced User Experience

Personalized Business Intelligence Profiling: The system add user profiles and adapt the depth of explanations and analysis based on their roles and level of expertise. This would simplify for different user types to understand and act on information through personalized conversational experiences.

8.3.3 Security and Compliance Enhancements

Enterprise Security Integration: Comprehensive security frameworks would address limitations with external API dependencies by adding features like end-to-end encryption, secure API gateways, and enterprise IAM integration. On-premises LLM deployment could be a viable solution that maintains information privacy while still allowing conversational AI capabilities [PML22].

Data Governance Framework: Compliance monitoring could enforce data policies, maintain audit trails, and generate transparency reports for GDPR and CCPA compliance.

8.3.4 Research Extensions

Cross-Domain Generalization: There are research opportunities to enable domain adaptation techniques allow system generalization in a wide range of business settings with minimal reconfiguration, which addresses the problem of domain-specific dataset dependencies.

These development opportunities include short-term practical extensions and long-term research opportunities that could improve conversational business intelligence while addressing the limitations found in the current work.

9 Conclusions

9.1 Summary of Contributions

This study addresses the gap in accessibility between advanced analytical tools and non-technical users who want insights on their data. The work makes three contributions that enhance both the theoretical understanding and the practical use of conversational analytics.

The theoretical contribution indicates that the trade-off between accessibility and rigor in business intelligence is not unavoidable. This study challenges ideas regarding that the usability and statistical sophistication are related by showing that complex statistical algorithms can be made simplified without losing their full analytical integrity.

The methodological contribution demonstrate the employment of prompt engineering to systematically transfer algorithmic knowledge. This method provides a way to encode domain knowledge into conversational AI systems which can be integrated with other advanced algorithms to make them conversationally accessible.

The practical contribution highlights that the idea is technically feasible by using a working system that achieves 100% statistical faithfulness and lets users use natural language to find advanced patterns. The comprehensive evaluation framework gives researchers a way for investigating at similar systems in the future.

9.2 Answers to Research Questions

The central research question examined how integration simplifies for non-technical users to access and act on information. The study shows that eliminating technical barriers greatly improves accessibility, allowing users to start complex analyses by asking questions in natural language. However, actionability is still limited by the fact that contextual processing can't handle long-term analytical conversations.

Full statistical preservation and reliable query classification prove that technical effectiveness is operational. User experience demonstrate that basic analytics support users to enhance understanding. Business value creation is partially achieved through improved decision speed, but full realization requires addressing contextual processing challenges.

9.3 Reflection on Research Objectives

The research process demonstrated both the potential and the challenges of using conversational analytics, which ultimately lead to question fundamental assumptions about AI-human collaboration in analytical contexts.

The most significant conclusion resulted from the fact that technical performance and user acceptance were different. Even though the technical metrics achieved high results, user confidence was only moderate. This shows a major flaw in how conversational analytics systems are perceived and judged This finding suggests that the research and system conducted has focused disproportionately on algorithmic performance while underestimating the psychological and organizational dimensions of analytical tool adoption.. For conversational analytics to work, future research needs to address user trust as a primary design constraint instead of a secondary one.

The contextual processing challenge was more fundamental than anticipated, revealing flaws in how large language model architectures work for long-term analytical reasoning.The limitation suggests that conversational analytics may require more than just simple prompt engineering or model scaling. They may require hybrid approaches that combine conversational interfaces with higher analytical memory systems.

This discovery has broader effects on AI-mediated analytics. It demonstrates that conversational AI is good at translation and increasing comprehension and understanding, however, is flawed at the long-term reasoning needed for complex analytical conversations. This limit may be what separates human and AI capabilities in analytical situations.

The research scope showed proof of concept while showing fundamental constraints that put this work within broader questions about the limits of current AI architectures for complex reasoning tasks.

9.4 Final Remarks

This research contributes to understanding how artificial intelligence can enhance human analytical capabilities while showing critical constraints that define current conversational analytics limits.

The difference between technical performance and user acceptance shows a fundamental challenge in AI system design: technical systems solely is insufficient for successful human-AI collaboration. This insight goes beyond conversational analytics and suggests that future AI systems must prioritize human dynamics and comprehension preference as primary design constraints rather than secondary considerations.

The research provides a systematic approach for making advanced algorithms conversationally accessible, however, the contextual processing limitations suggest each domain may face different implementation constraints. Instead of a unified technological solution, conversational analytics may develop as domain-specific approach tailored to particular analytical challenges.

Looking forward, the optimal future steps for this research may involve integration of conversational interfaces with traditional analytical tools leveraging AI for accessibility enhancement while preserving human expertise for complex reasoning. This hybrid model may represent the most effective approach until advances in AI reasoning capabilities emerge, ultimately enabling truly collaborative and fully effective analytical partnerships between humans and artificial intelligence.

References

[A+20] Alejandro Barredo Arrieta et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[AFKS24] Martin Atzmueller, Johannes Fürnkranz, Tomáš Kliegr, and Ute Schmid. Explainable and interpretable machine learning and data mining. *Data Mining and Knowledge Discovery*, 38:2571–2595, July 2024.

[Amj23] Asad Amjad. Conveying complex data insights to non-technical management. *LinkedIn*, 2023. Available at https://www.linkedin.com/pulse/conveying-complex-data-insights-non-technical-management-asad-amjad-2ghdf/.

[Ana23] Number Analytics. 8 surprising analytics stats boosting data-driven decision making. https://www.numberanalytics.com/blog/8-surprising-analytics-stats-boosting-data-driven-decision-making, 2023. Accessed: 2025-06-19.

[Atz15] Martin Atzmueller. Subgroup discovery - advanced review. *WIREs Data Mining and Knowledge Discovery*, 2015. Available at https://www.kde.cs.uni-kassel.de/wp-content/uploads/atzmueller/paper/atzmueller-subgroup-discovery-advanced-review-wires-2015.pdf.

[AY+24] Muhammad Ahmad, Muhammad Younas, et al. Microservices vs monolith: A comparative analysis and problem-solving approach in web development area. In *2024 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, pages 1–6. IEEE, 2024.

[BHH+24] Nimrod Busany, Ethan Hadar, Hananel Hadad, Gil Rosenblum, Zofia Mazzlanka, Okhaide Akhigbe, and Daniel Amyot. Automating business intelligence requirements with generative ai and semantic search. *arXiv*, December 2024.

[BV22] Bernadette Bensaude-Vincent. Philosophy of science and the posthuman condition. *Journal of Posthumanism*, 2(1):27–37, 2022.

[CE11] C. J. Carmona and D. Elizondo. Subgroup discovery: Real-world applications. Technical report, University of Burgos and De Montfort University, 2011. Available at https://simidat.ujaen.es/sites/default/files/2024-05/TR2011.pdf.

[Dah24] Fahad Yahya M Dahish. The impact of conversational ai on business intelligence transforming data interaction and decision-making. *Journal for Research on Business and Social Science*, 7(8), 2024. ISSN (Online) 2209-7880.

[DBZB24] E. Derner, K. Batistic, J. Zahalka, and R. Babuska. A security risk taxonomy for prompt-based interaction with large language models. *IEEE Access*, 12:126176–126187, 2024.

[DLL+17] Nicola Dragoni, Ivan Lanese, Søren Larsen, Manuel Mazzara, Ruslan Mustafin, and Ana Safina. Microservices: Yesterday, today, and tomorrow. *Present and Ulterior Software Engineering*, pages 195–216, 2017.

[DVK17] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[EA22] Nada Salaheddin Elgheriani and Nuredin D Ali Salem Ahme. Microservices vs. monolithic architectures: The differential structure between two architectures. *MINAR International Journal of Applied Sciences and Technology*, 4, 2022.

[Ela25] Elastic. What are large language models? elastic.co, 2025. Available at https://www.elastic.co/what-is/large-language-models.

[FRF+02] Martin Fowler, Dave Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, and Randy Stafford. *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, Boston, MA, 2002.

[GHJV94] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. Addison-Wesley Professional, Boston, MA, 1994.

[Her11] F. Herrera. Statistical inference in computational intelligence and data mining. *Knowledge and Information Systems*, 2011. Available at https://sci2s.ugr.es/sites/default/files/ficherosPublicaciones/1324_2011-Herrera-KAIS.pdf.

[HHR+23] Peter Henderson, Haoran Hu, Jeffrey Romoff, Emma Brunskill, Joelle Pineau, and David Meger. Towards robust and reliable large language models: Challenges and opportunities. In *Proceedings of the 2023 Conference on Neural Information Processing Systems (NeurIPS 2023) Workshop on Responsible AI*, 2023. Preprint available on arXiv.

[KCFS08] Arno Knobbe, Bruno Crémilleux, Johannes Fürnkranz, and Martin Scholz. From local patterns to global models: The lego approach to data mining. In *Proceedings of the International Workshop on Local Pattern Detection*, 2008.

[Keb22] Keboola. 5 stats that show how data-driven organizations outperform their competition. https://www.keboola.com/blog/5-stats-that-show-how-data-driven-organizations-outperform-their-competition, 2022. Accessed: 2025-06-19.

[Klö92] Willi Klösgen. Problems for knowledge discovery in databases and their treatment in the statistics interpreter explora. *International Journal of Intelligent Systems*, 7(7):649–673, 1992.

[LJ14] Fei Li and H. V. Jagadish. Constructing natural language interfaces to sql databases using reinforcement learning. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 73–84. ACM, 2014.

[LYF+23] Pengchuan Liu, Weizhu Yuan, Jinpeng Fu, Zheng Jiang, Hiroaki Hayashi, and Graham Neubig. Learning to prompt for vision-language models. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, 2023.

[MH25] Shinya Masadome and Taku Harada. Reward design using large language models for natural language explanation of reinforcement learning agent actions. *IEEJ Transactions on Electrical and Electronic Engineering*, March 2025.

[MK11] M Meeng and A Knobbe. Flexible enrichment with cortana–software demo. In *Proceedings of BeneLearn*, 2011.

[OIA+24] S. O. Ogundoyin, M. Ikram, H. J. Asghar, B. Z. H. Zhao, and D. Kaafar. A large-scale empirical analysis of custom gpts' vulnerabilities in the openai ecosystem. *arXiv preprint arXiv:2402.09631*, 2024.

[Ope23] OpenAI. Gpt-4 technical report, 2023. Available at: https://openai.com/research/gpt-4.

[Pal23] Willem Jan Palenstijn. pysubdisc: Python wrapper for subdisc: Subgroup discovery, 2023. Version 0.1.0.

[PML22] Elena Perez, Jie Mu, and Mike Lewis. Personalized conversational agents for business intelligence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1245, 2022.

[Res25] Meticulous Research. Business intelligence market poised to reach $56.9 billion by 2032 – strategic market analysis report. *finance.yahoo.com*, March 2025.

[Unk25] Unknown. You're facing pushback from non-technical team members. *LinkedIn Advice*, 2025. Available at https://www.linkedin.com/advice/1/youre-facing-pushback-from-non-technical-team-members-j30ff.

[Zho20] Zhi-Hua Zhou. Adaptive learning for classification with noisy labels. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2413–2426, 2020.