

## **Problem Statement**

In my current role, I regularly build ServiceNow dashboards that visualize IT Service Management (ITSM) metrics such as incident volume, service requests, and customer interactions. These dashboards offer valuable insights for tracking key performance indicators (KPIs) and serve as a reference for analysts who craft weekly or monthly reports for executives, managers, and team members. However, most of the reporting process remains manual and time-consuming. Executives and managers rely on these reports to understand trends, identify performance issues, and make informed decisions, yet generating them requires repetitive data exploration, formula validation, and narrative writing.

This project aims to automate insights and analytics from ServiceNow data. By analyzing incident, request, and interaction records, the system will produce audience-specific summaries that highlight key trends, forecasts, and actionable recommendations. Users will have a traceability feature to understand how metrics were calculated or view underlying records. Additionally, a chatbot interface will allow users to ask questions about metrics, trends, data definitions, and business rules based on the uploaded data.

Solving this problem is both relevant and impactful, as it combines machine learning, natural language processing, and data analysis to support business decisions across organizational levels. Automation increases analysts' productivity while improving consistency, transparency, and quality.

## **Data Description and Acquisition**

For this project, a representative subset of ServiceNow data will be used, consisting of approximately 20,000 to 35,000 records from each of the three tables—incidents, requests, and interactions (about 85,000 records in total). Each dataset includes 10 to 20 fields, such as unique identifiers, requester and assignee information, states, priority levels, communication channels, timestamps, and activity metrics like reopen and reassignment counts. The datasets span the past one to two years, providing sufficient historical context to support trend detection, forecasting, and insight generation. Data extraction will be performed in CSV or Excel format, consistent with the formats used in current dashboard workflows. To protect privacy and confidentiality, all personally identifiable information and sensitive business data will be anonymized, randomized, and/or timestamp-shifted, preserving realistic operational patterns without exposing actual records.

Before analysis, the data will be preprocessed including cleansing, normalization, and feature engineering. These structured datasets will have high quality, making it effective for statistical analysis, machine learning, and natural language processing. The size is manageable which allows for the creation of a working prototype with capability to demonstrate operational complexity and meaningful insights.

## **Proposed Approach**

A structured workflow will be followed including data preprocessing, insight extraction, summary generation, and chatbot interaction. ServiceNow data from incidents, requests, and interactions will be cleansed, normalized, and merged into a unified dataset. Key metrics such as closure rates, average resolution times, and reassignment frequencies will be derived for trend detection and analysis. Outliers and missing values will be handled using standard statistical methods (e.g., IQR, imputation). The analysis will use a hybrid approach, combining traditional statistical methods with lightweight machine learning. Supervised models (regression) will forecast metrics like incident volume, while unsupervised methods (clustering) will detect anomalies and categorize data. Predictors will include time-based features, priority, categories, and assignment groups. After extracting insights, rule-based templates and language models will generate human-readable summaries that will highlight KPIs, trends, and

actionable recommendations for each tailored audience—executives, managers, and team members. A traceability feature will allow users to view the underlying records or calculation logic. Additionally, a chatbot powered by a retrieval-augmented generation (RAG) approach will answer user queries (e.g., "Which teams have the highest close rates?"). The chatbot will retrieve relevant data from the uploaded datasets and provide context-aware, data-driven answers.

The prototype will be built using Python, with libraries like pandas and scikit-learn for data processing and analysis, and Streamlit or Flask for the web-based interface. Natural language processing will be handled via the Hugging Face API or similar models, with all data processing occurring locally to ensure security. This approach strikes a balance between automation, interpretability, and scalability to deliver a functional, practical system.

## **Deliverables**

The final deliverable will be a working prototype showcasing automated AI-insights for ServiceNow data, deployed as a web-based application accessible via a browser. Users can interact with the system through an intuitive interface, enabling interactive conversation for reporting data and insights.

Key components include:

- **Dataset Upload:** Users can upload CSV or Excel files containing incident, request, and interaction data.
- **Automated Summaries:** Generation of “email-ready,” audience-specific reports highlighting key trends, forecasts, and actionable recommendations.
- **Traceability:** Each metric can be explored to reveal its underlying records or calculation logic.
- **Interactive Chatbot:** A conversational assistant will answer questions about metrics, trends, and calculations based on the uploaded datasets.

The prototype is designed for scalability, allowing future enhancements such as customizable summary content, more advanced predictive models, or direct integration with ServiceNow.

## **Computational Resources**

Development and testing of this project will be conducted on a personal workstation equipped with an Intel Core i7-8550U processor (4 cores, 8 threads, 1.8 GHz base frequency) and 16 GB of RAM. This configuration provides adequate processing power and memory for the project's scope, which involves working with structured datasets of approximately 80,000 to 90,000 records. Tasks such as data preprocessing, statistical analysis, and natural language processing using lightweight machine learning models can be efficiently executed on this setup. Since the system primarily relies on data manipulation, summarization, and inference from pre-trained language models accessed through APIs, GPU acceleration is not required for the initial prototype. Traditional machine learning and rule-based summarization tasks can be fully performed using CPU resources. If additional computational capacity is needed, cloud-based environments such as Google Colab or AWS can be leveraged to provide temporary GPU or higher-memory instances.