

## **Industry Project Outline**

### **Triage System for Support Tickets Using Large Language Models (LLMs)**

#### **Background**

Datel has accumulated a vast support database over 30 years across a large portfolio of software applications. Despite this wealth of information, our support teams often rely on personal expertise when addressing support cases, with some tickets only being resolvable by specific personnel. This is particularly critical for one of our products which is nearing its end of support. As experienced staff may move to other products or reduce working hours, Datel needs to continue providing high-quality support to our customers while potentially operating with fewer resources.

The goal of this project is to leverage the extensive information in our support database to automatically triage support tickets when they are logged and provide the support team with potential solutions, complete with references and links to similar cases. This should help to drive growth as it will allow us to be as effective, if not more effective, in solving support requests and meeting the growing demand on these teams without the need for additional personnel. In the future, this system could offer initial self-help guidance to customers; however, this feature would need to account for data security, ensuring that sensitive information from unrelated tickets is not exposed.

#### **Database Overview**

Our support database comprises structured and unstructured data stored in various formats:

Structured Data: Each ticket includes key fields such as:

1. Product affected
2. Specific area of the product (e.g., Accounts Payable)
3. Business impact (e.g., business-critical, standard issue)
4. Key dates (received, assigned, completed)
5. Support consultant assigned
6. Ticket priority level
7. Simple summary and detailed textual description (e.g., "Issue with unallocated payment")

Unstructured Data: This includes:

Detailed ticket descriptions: Free-text fields provided by the client, often including problem descriptions, context, and any troubleshooting steps taken.

Attachments: Files such as emails, screenshots, or additional documentation, stored either as database blobs or in file shares.

Ticket history: Including internal notes, email exchanges with the client, and any other communications, all stored in a relational database table.

## Goals

1. **Data Improvement Analysis:** Document and identify any improvements in the structure and organization of our support data that could enhance the performance of a triaging system.
2. **LLM Exploration:** Explore the potential of pre-trained Large Language Models (e.g., GPT) to extract meaningful insights from both structured and unstructured data, and map these insights to new ticket requests.
3. **Triage System Design:** Develop a triage system that:  
Automatically classifies support requests based on ticket data and descriptions.  
Identifies potential solutions or troubleshooting steps from historical support data.  
Prioritizes tickets based on business impact and urgency.  
Recommends appropriate personnel or teams to handle specific tickets.
4. **Prototype Implementation:** Create a working prototype that:  
Leverages pre-trained LLMs to integrate both structured and unstructured data.  
Provides suggested resolutions for the support team based on historical tickets.  
Proposes priority-based ticket assignments to available resources.
5. **Scalability Consideration:** Discuss potential future extensions, including providing self-service capabilities to customers, with careful attention to data security and client confidentiality.

## Deliverables

1. **Literature Review:** A detailed overview of models and techniques used in similar systems for automated triage and support ticket management.

2. **Data Processing Pipeline:** Steps for preparing, cleaning, and processing both structured and unstructured data for model consumption.
3. **Prototype System:** A working prototype that integrates the selected LLM(s) and demonstrates triaging, prioritization, and solution identification.
4. **Evaluation Metrics:** A clear set of evaluation criteria to measure the prototype's effectiveness (e.g., classification accuracy, solution relevance, response time improvements).
5. **Final Report:** A comprehensive project report covering all aspects of the project, including findings, system architecture, implementation details, and recommendations for future development.

## Requirements

1. Strong understanding of relational databases and unstructured data processing.
2. Excellent data analysis skills to identify potential improvements in the current data schema and structure.
3. Experience with Large Language Models and natural language processing (NLP) techniques, including model fine-tuning and prompt engineering.
4. Knowledge of programming languages and familiarity with relevant machine learning libraries
5. An understanding of software support processes, especially in handling tickets, prioritization, and resource allocation.
6. Familiarity with data security concerns, especially in the context of sensitive client data and confidentiality.

## Additional Considerations

**Ethics and Privacy:** Given that customer tickets may contain sensitive information, outline how the prototype will ensure the privacy and confidentiality of the data it processes (e.g., masking techniques, access control).

**Computational Resources:** Include a note on computational requirements, such as the need for cloud resources, GPUs, or other infrastructure to process large datasets or fine-tune LLMs.

**Team Collaboration:** Define any expectations regarding how the project team will collaborate with support staff, IT, or other internal teams to gather feedback on the prototype's effectiveness.

## Project Summary for MSc Data Science Students

### 1. Structure of the Project

The project is divided into the following key phases:

1. **Research & Data Understanding** – Understanding the current support ticket database and identifying areas for improvement with regards to tagging and data classification.
2. **Exploration of LMs** – Investigating the use of pre-trained Language Models (e.g., GPT) to extract insights from structured and unstructured data, or the creation of new models. Also determine the data and human intervention needed to train the system.
3. **Triage System Design** – Defining a system to classify and prioritize tickets, suggest solutions, and recommend assignments. Consider how the system performance will be measured. Also consider ongoing training of the system.
4. **Prototype Development** – Implementing a working prototype that demonstrates the triage system.
5. **Evaluation & Future Considerations** – Measuring system performance and suggesting future improvements. Identify how we might test for successful results – how do we compare with human classification of tickets? How can we measure this?

### 2. Objectives

The project aims to:

- Improve **data organisation** for better triaging performance.
- Utilize **LMs to classify and extract insights** from support tickets.
- Develop a **triage system** that automates classification, prioritization, and solution recommendations.
- Implement a **prototype** showcasing the system's effectiveness.
- Ensure **data privacy and security** when handling customer information.
- Explore **scalability** for potential customer self-service capabilities.

### 3. Tasks Assigned to Students

The six students will be assigned tasks across different phases of the project:

#### Phase 1: Research & Data Understanding

- **Task 1:** Conduct a review on automated triage systems using AI/ML.

- **Task 2:** Analyse Datel's structured and unstructured support ticket data.
- **Task 3:** Discuss the use of Datel's support system from tickets being logged through to resolution.
- **Task 4:** Identify data improvement opportunities to enhance model accuracy, including data tagging.

## **Phase 2: LM Exploration**

- **Task 5:** Investigate and compare various LMs for ticket classification.
- **Task 6:** Experiment with fine-tuning or prompt engineering for extracting meaningful insights.

## **Phase 3: Triage System Design**

- **Task 7:** Develop a model pipeline to:
  - Automatically classify support tickets.
  - Suggest potential solutions using historical ticket data.
  - Prioritize tickets based on urgency and business impact.
  - Recommend the appropriate team or personnel for handling cases.

## **Phase 4: Prototype Development**

- **Task 8:** Implement a prototype integrating structured and unstructured data.
- **Task 9:** Develop a mechanism for validating model recommendations.

## **Phase 5: Evaluation & Future Considerations**

- **Task 10:** Define key performance metrics (e.g., classification accuracy, resolution speed).
- **Task 11:** Assess potential risks, such as **data security & confidentiality** concerns.
- **Task 12:** Explore computational resource needs (e.g., GPU/cloud computing).
- **Task 13:** Document findings and prepare a final project report with recommendations.

## **4. Additional Considerations for Industry Project**

- **Ethical Considerations:** Ensure ticket data privacy (e.g., anonymization techniques).

- **Stakeholder Collaboration:** Engage with Datel's support team for real-world feedback.
- **Computational Resources:** Identify infrastructure requirements for model training/deployment.
- **Scalability Planning:** Consider future expansions like self-service ticketing systems.

## 5. Requirements from Datel for the Project

- **Access to Data:** A sample of historical support tickets (structured & unstructured).
- **Regular Check-ins:** Weekly/bi-weekly meetings for project alignment.
- **Stakeholder Feedback:** Insights from Datel's support team to refine model recommendations.
- **Final Evaluation:** Assessment of prototype effectiveness and feasibility for real-world deployment.