Utility of the prototype

This prototype supports hotel managers by predicting booking cancellation risk and recommending appropriate actions. Beyond the initial machine learning model, this version integrates a Large Language Model (LLM) to generate human-readable summaries based on prediction results and booking details. Users can toggle between rule-based and AI-generated advice, and the entire result (including the LLM output) is processed and included in a downloadable PDF report. This layered architecture, combining statistical modelling, prompt engineering, and AI post-processing, adds functional and technical depth to the user experience.

Main Design Decisions Chosen

The prototype was designed with clarity and usability in mind. The user interface was built using Streamlit's column layout to separate user input from prediction output, keeping the experience intuitive. We introduced a toggleable choice between rule-based advice and AI-generated summaries to ensure user control and transparency. The LLM is prompted dynamically using structured model outputs and user input features (such as lead time and special requests), the output of the LLM is sanitized for compatibility with the PDF export. Additionally, the PDF generation was integrated as a seamless, downloadable report containing tailored insights, ensuring that the tool provides value beyond real-time display. The decision to include the LLM summary only when selected, both visually and in the PDF, reinforced a dynamic, user-centered design.

Main Difficulties Found

During the development, two main technical challenges were encountered. The first was ensuring compatibility between the LLM output and PDF generation. The fpdf library requires Latin-1 encoding, which conflicted with the character set returned by the LLM. To resolve this, a sanitization step was implemented to normalize the text and safely encode it before export. The second major difficulty was adapting to changes in the OpenAI Python library. The latest version introduced a new syntax that did not match our previous implementation. This required updating our integration logic and securely managing the API key using environment variables. In addition to these, we faced a few smaller design-related challenges, which we were able to solve with some conditional code parts.

Link to Streamlit demonstration

Recording-20250324_144513.webm