SOLUTION APPROACH

The objective of the project is to create a Python app that uses a LLM model, like llama – 3B, to read CSV files, produce graphs, respond to queries, and analyse data statistically. The project follows the flow :– Reading and parsing CSV files, executing fundamental statistical analysis (mean, median, mode, standard deviation, and correlation coefficients), creating a variety of plots (histograms, line plots, scatter plots, etc.) specifically based on the user's query, and comprehensively responding to user inquiries.

The data is transformed into a format that the LLM can easily process in order for it to be able to comprehend and analyse. Tokenizing the text data and creating numerical embeddings – in vector form (that represent the semantically precise meaning, is a proficient way to approach the problem statement and thus allows us to provide the model, a base for performing the analysis and statistics efficiently). This can be utilized by various libraries that python offers.

Based on the input tokens and their embeddings, the LLM learns during training to predict the subsequent token in a sequence. This enabled in producing text which can be easily comprehended by us in our daily language further allowing us in responding to queries in light of the context that is given.

Step by step breakdown of code-

Installing the necessary modules and libraries- langchain, langchain_community, pandasai, ollama, streamlit, groq, langchain-groq, python-dotenv

Langchain (library) – NLP tasks like tokenization and for embedding generation

langchain_community (module) – for utilizing the essential extended modules contributed by the langchain community

Pandasai (module or extension of pandas) – extends pandas library for data analysis, also incorporating the automated insights about data using ai driven features and data handling

Ollama (library) – Offers charting and data visualisation tools and utilities

Streamlit (library) – development of interactive web apps for data analysis and visualisation

Python-dotenv (module) – Securely maintains environment variables, guaranteeing that private data, like API keys, is handled properly when configuring applications.

Groq – Groq is a query language that optimises performance and flexibility while querying massive datasets

The queries_with_model function allows data analysis based on user queries collected from user_csvs by integrating the groq_api_key from groqcloud to connect with the llama-3B model. The pandas_ai set up in the SmartDataFrame effectively evaluates and examines user questions efficiently utilizing the LLM model, using data from CSV files for in-depth examination.

The variable graph_types is managed by st.sidebar. The graph plotting feature in Streamlit is controlled by a selectbox that provides users with a toggle menu of visualisation options. For the aim of in-depth data analysis, the LLM model further improves these visualisations. Robust error management tools, such as pre-programmed alert messages, improve user experience by alerting users to missing values or misspelt words in a timely manner.