Project Name: Semantic Search Application using Transformers and KNN

1. Brief Explanation of Project:

To build a tool that allows users to intelligently search for offers via text input from the user.

Data Preparation: Loaded and explored datasets using popular Python libraries like Pandas and PySpark. The tool allowed me to understand data structure, check for missing values, encode categorical variables, and perform efficient preprocessing.

Semantic Search: The application integrates Sentence Transformers and KNN for semantic search. Users can search for similar offers based on text embeddings. Two different approaches are showcased: one using Faiss for K-nearest neighbor search and the other using scikit-learn' s Nearest Neighbors for contextual search.

Assumptions:

1. The data preprocessing steps enhance data quality for analysis and search.

2. Semantic search using advanced models will yield meaningful and relevant results.

3. The semantic search function can intelligently determine the type of input text.

Trade-offs:

1. **Data Quality vs. Computational Cost:** Intensive preprocessing ensures high data quality but demands significant computational resources

2. **Speed vs. User-Friendliness**: Faiss offers speed in similarity search, while scikit-learn prioritizes user-friendliness, exemplifying the trade-off between efficiency and ease of use.

Conclusion:

Our project was initiated by configuring the necessary models, such as Nearest Neighbors and Sentence Transformers, to enable semantic search and text embedding capabilities.

Three primary functions were defined: data engineering techniques, Faiss-based semantic search, and scikit-learn-based contextual search.

The user interface was built using Gradio, providing a user-friendly, interface for performing semantic search functionalities to output related OFFER's based on the search query. The entire application was developed using Python and tested using unit testing techniques.