# **Text-to-SQL System Documentation**

## **Overview**

The Text-to-SQL system is an AI-powered pipeline that converts natural language queries into SQL queries, retrieves relevant information from structured (CSV) and unstructured (PDF) data, and generates visualizations based on the results. The system leverages AWS Bedrock's Claude 3.5 Sonnet for natural language understanding and FAISS for efficient vector-based retrieval.

## **Features**

### **1. Text-to-SQL Conversion**

* Converts user queries into SQL queries based on the given CSV dataset.
* Uses both structured (CSV) and unstructured (PDF) data for context-aware SQL generation.
* Leverages Claude 3.5 Sonnet via AWS Bedrock for accurate SQL generation.

### **2. Content Retrieval**

* Retrieves relevant data from CSV and PDF files using FAISS vector search.
* Supports efficient semantic search for both structured and unstructured data.

### **3. Visualization Generation**

* Automatically generates visualizations using Vega-Lite JSON specifications.
* Supports different chart types like bar charts and pie charts.
* Saves visualization as a PNG file.

## **Code Approach**

### **1. Setup and Initialization**

* Loads AWS credentials using boto3.Session.
* Initializes the Bedrock model for embedding generation.

### **2. Data Loading and Processing**

* Loads CSV data using CSVLoader and splits it into chunks using CharacterTextSplitter.
* Loads PDF documents using PyMuPDFLoader without requiring OCR.
* Assigns metadata to distinguish between CSV and PDF sources.

### **3. FAISS Vector Store**

* Creates FAISS-based vector storage for efficient retrieval.
* Generates embeddings using Bedrock and stores indexed data.
* Supports loading previously saved FAISS indices for performance optimization.

### **4. Query Execution**

* Retrieves relevant context from CSV and PDF files.
* Calls AWS Bedrock to generate an SQL query based on the retrieved context.
* Parses JSON output to execute SQL and extract meaningful insights.

### **5. Visualization Generation**

* Uses Altair and Vega-Lite to generate visualizations based on query results.
* Converts the Vega-Lite JSON specification into a PNG file.
* Saves the generated visualization dynamically.

## **Usage**

if \_\_name\_\_ == "\_\_main\_\_":

text2sql = TextToSQL(csv\_path="data/data.csv", pdf\_path="data/data.pdf", faiss\_index\_path="faiss\_index")

query = "Fetch the GPA of id number 1141"

sql\_query = text2sql.generate\_sql\_query(query)

print(sql\_query)

## **Dependencies**

* boto3: AWS SDK for Python
* langchain: For text processing and vector-based retrieval
* FAISS: For efficient similarity search
* altair: For visualization generation
* vl-convert-python: For converting Vega-Lite JSON to PNG

Claude 3.5 Pricing Details

### **1. Claude 3.5 Sonnet**

* **Price per 1,000 input tokens**: $0.003
* **Price per 1,000 output tokens**: $0.015
* **Batch pricing per 1,000 input tokens**: $0.0015
* **Batch pricing per 1,000 output tokens**: $0.0075
* **Cache write pricing per 1,000 input tokens**: $0.00375
* **Cache read pricing per 1,000 input tokens**: $0.0003

### **2. Claude 3.5 Haiku**

* **Price per 1,000 input tokens**: $0.0008
* **Price per 1,000 output tokens**: $0.004
* **Batch pricing per 1,000 input tokens**: $0.0005
* **Batch pricing per 1,000 output tokens**: $0.0025
* **Cache write pricing per 1,000 input tokens**: $0.001
* **Cache read pricing per 1,000 input tokens**: $0.00008

## **Conclusion**

This Text-to-SQL system provides an automated way to generate SQL queries from natural language, retrieve relevant data from structured and unstructured sources, and generate insights through visualizations. It is a scalable and efficient approach for data analysis and querying.