This app will include the data analysis features from Project 1, the RAG system from Project 2, and the conversational AI from Project 3.

This combined app now includes:

1. Data Overview: Displays basic statistics and distributions from the uploaded CSV file.
2. Detailed Analysis: Provides in-depth analysis including sentiment analysis, correlations, and visualizations.
3. Query System: Allows users to ask questions about the customer feedback using the RAG system.
4. Sentiment Analysis: Performs sentiment analysis on the comments using a local model and displays the results.

The app uses a combination of NLTK's VADER for quick sentiment analysis in the Detailed Analysis section and a more advanced transformer-based model for the dedicated Sentiment Analysis page. It also incorporates the RAG system for answering user queries about the feedback data.

Users can navigate between these different analyses using the sidebar, providing a comprehensive tool for analyzing customer feedback data.

# Detailed Notes: Customer Feedback Analysis Project

## 1. Project Overview

The Customer Feedback Analysis project is a comprehensive Streamlit-based web application designed to process, analyze, and derive insights from customer feedback data. It combines various data analysis techniques, natural language processing, and machine learning to provide a multi-faceted view of customer sentiments and satisfaction levels.

## 2. Core Components and Features

### 2.1 Data Loading and Preprocessing

* Function: load\_data(file)
* Capabilities:
  + Reads CSV files using pandas
  + Handles large datasets by sampling (threshold: 10,000 rows)
  + Converts date columns to datetime format
  + Calculates resolution time in hours

### 2.2 Vector Store Creation

* Function: create\_vector\_store(df)
* Purpose: Enables efficient similarity search for the query system
* Process:
  + Uses RecursiveCharacterTextSplitter for text chunking
  + Employs GoogleGenerativeAIEmbeddings for creating embeddings
  + Implements caching mechanism to store and retrieve embeddings
  + Creates a FAISS index for fast similarity search
* Performance Optimization:
  + Implements batch processing with a progress bar
  + Caches results to disk for faster subsequent loads

### 2.3 LLM Initialization

* Function: initialize\_llm()
* Utilizes ChatGoogleGenerativeAI with the Gemini 1.5 Pro model
* Configured for deterministic output (temperature = 0)

### 2.4 Data Overview Page

* Function: data\_overview(df)
* Features:
  + Displays raw data in a Streamlit dataframe
  + Shows basic statistical summary
  + Visualizes service satisfaction distribution (histogram)
  + Presents requests by company distribution (pie chart)

### 2.5 Detailed Analysis Page

* Function: detailed\_analysis(df)
* Analyses:
  + Sentiment analysis using NLTK's VADER
  + Correlation between service satisfaction and sentiment scores
  + Company-wise analysis of satisfaction, sentiment, and resolution time
  + Resolution time analysis across companies
  + Word cloud generation from customer comments

### 2.6 Query System Page

* Function: query\_system(vector\_store, llm)
* Features:
  + Implements a conversational retrieval chain
  + Uses ConversationBufferMemory for maintaining chat history
  + Allows users to input questions about the feedback data
  + Generates AI-powered responses based on the dataset
  + Displays chat history in the sidebar

### 2.7 Sentiment Analysis Page

* Function: sentiment\_analysis(df)
* Capabilities:
  + Uses a pre-trained transformer model for sentiment analysis
  + Provides a sentiment summary of all comments
  + Visualizes sentiment distribution
  + Displays average satisfaction score
  + Shows sample comments with their sentiment scores

## 3. Technologies and Libraries Used

### 3.1 Core Technologies

* Streamlit: For creating the web application interface
* Pandas: For data manipulation and analysis
* NLTK: For basic sentiment analysis (VADER)
* Hugging Face Transformers: For advanced sentiment analysis
* Plotly Express: For interactive data visualizations
* Matplotlib: For generating the word cloud
* LangChain: For building the conversational AI system
* FAISS: For efficient similarity search in high-dimensional spaces

### 3.2 AI and NLP Components

* Google Generative AI (Gemini model): For powering the query system
* GoogleGenerativeAIEmbeddings: For creating text embeddings
* SentimentIntensityAnalyzer (NLTK): For quick sentiment analysis
* Pre-trained sentiment analysis model: For in-depth sentiment analysis

### 3.3 Data Visualization

* Plotly Express: Used for creating interactive charts (histograms, scatter plots, bar charts)
* Matplotlib: Employed for generating the word cloud visualization

## 4. Application Flow

1. User uploads a CSV file containing customer feedback data
2. The app processes the data and creates a vector store
3. Users can navigate between different analysis pages using the sidebar
4. Each page provides specific insights and visualizations
5. The query system allows for interactive exploration of the data

## 5. Key Features and Insights Provided

* Distribution of service satisfaction ratings
* Sentiment analysis of customer comments
* Correlation between satisfaction ratings and sentiment scores
* Company-wise performance analysis
* Resolution time analysis across different companies
* Word frequency visualization through word cloud
* AI-powered query system for exploring the dataset
* Detailed sentiment analysis with score distribution

## 6. Performance Considerations

* Implements data sampling for large datasets
* Uses caching mechanisms for expensive operations (data loading, vector store creation)
* Employs batch processing for embedding creation

## 7. Potential Enhancements

* Implement more advanced NLP techniques (e.g., topic modeling, named entity recognition)
* Add predictive analytics capabilities (e.g., predicting future satisfaction scores)
* Enhance the query system with more context-aware and personalized responses
* Optimize for even larger datasets with distributed processing
* Implement user authentication and data privacy features
* Add export functionality for analysis results and visualizations

## 8. Conclusion

The Customer Feedback Analysis project provides a robust, user-friendly interface for businesses to gain valuable insights from their customer feedback data. By combining traditional data analysis techniques with advanced NLP and AI capabilities, it offers a comprehensive toolkit for understanding customer sentiments, identifying areas for improvement, and making data-driven decisions to enhance customer satisfaction.