🧠 AI Analytics - Intelligent Data Analysis Platform

A powerful Streamlit web application that combines the simplicity of natural language with advanced AI capabilities to perform comprehensive data analytics on CSV files. Built with PandasAI and Google's Generative AI, this platform makes data analysis accessible to everyone.

🎈 STREAMLIT   **🐍 PYTHON**   **🐳 DOCKER**

🎯 Features

- 🔍 **Natural Language Queries**: Ask questions about your data in plain English
- 📊 **Interactive Data Visualization**: Generate charts and graphs automatically
- 🤖 **AI-Powered Insights**: Leverage Google's PaLM AI for intelligent data analysis
- 🗂 **CSV File Support**: Easy drag-and-drop file upload
- 🔲 **Statistical Analysis**: Perform complex statistical operations with simple queries
- 🎨 **Beautiful UI**: Clean, intuitive Streamlit interface
- 🐳 **Docker Ready**: Containerized for easy deployment

🎥 Demo

YouTube Tutorial: Watch the complete walkthrough

Live Demo: Try it now on Streamlit Cloud

🚀 Quick Start

Prerequisites

- Python 3.8 or higher
- Google PaLM API key

Installation

1. **Clone the repository**

```
git clone https://github.com/Yash-Kavaiya/ai-analytics.git
cd ai-analytics
```

2. **Install dependencies**

```
pip install -r requirements.txt
```

3. **Set up your Google PaLM API key**

   ⚠️ **Security Note**: Never commit API keys to version control!

- Get your API key from Google AI Studio
- Replace the hardcoded API key in `streamlit_app.py` with your own:

```
llm = GooglePalm(api_key="YOUR_API_KEY_HERE")
```

For production, use environment variables:

```
import os
llm = GooglePalm(api_key=os.getenv("GOOGLE_PALM_API_KEY"))
```

4. **Run the application**

```
streamlit run streamlit_app.py
```

5. **Open your browser** and navigate to `http://localhost:8501`

🐳 Docker Deployment

1. **Build the Docker image**

```
docker build -t ai-analytics .
```

2. **Run the container**

```
docker run -p 8080:8080 ai-analytics
```

3. **Access the app** at `http://localhost:8080`

📖 Usage Guide

1. **Upload Data**: Click "Upload a CSV file" and select your dataset
2. **View Data**: The uploaded data will be displayed in a table format
3. **Ask Questions**: Type natural language questions about your data, such as:
   - "What is the mean and standard deviation of the glucose column?"
   - "Show me a correlation matrix"
   - "Create a histogram of age distribution"
   - "What are the top 5 values in the sales column?"
4. **Get Results**: Click "Analyze" to get AI-powered insights and visualizations

Example Queries

```
- "What is the average age of patients with diabetes?"
- "Show me a scatter plot of glucose vs insulin"
- "Calculate the correlation between all numeric columns"
- "What is the distribution of the outcome variable?"
- "Find outliers in the BMI column"
```

📁 Project Structure

```
ai-analytics/
├── streamlit_app.py          # Main Streamlit application
├── requirements.txt          # Python dependencies
├── Dockerfile                # Docker configuration
├── diabetes.csv              # Sample dataset
├── pandasai.ipynb            # Jupyter notebook for experimentation
├── pandasai.log              # AI conversation logs
├── cache/                    # Cached AI responses
├── marketing youtube thumbnail.png  # Project thumbnail
└── README.md                 # This file
```

🛠️ Technologies Used

- **Streamlit**: Web application framework
- **PandasAI**: AI-powered data analysis
- **Google PaLM**: Large language model
- **Pandas**: Data manipulation and analysis
- **Matplotlib**: Data visualization
- **Docker**: Containerization

🔧 Configuration

Environment Variables

For production deployment, consider using environment variables:

```
export GOOGLE_PALM_API_KEY="your_api_key_here"
export STREAMLIT_SERVER_PORT=8501
export STREAMLIT_SERVER_ADDRESS=0.0.0.0
```

Streamlit Configuration

You can customize the app by modifying the page configuration in `streamlit_app.py`:

```
st.set_page_config(
    page_title="Your Custom Title",
    layout="wide",
```

```
        initial_sidebar_state="expanded"
    )
```

🤝 Contributing

Contributions are welcome! Please feel free to submit a Pull Request. For major changes, please open an issue first to discuss what you would like to change.

1. Fork the repository
2. Create your feature branch (`git checkout -b feature/AmazingFeature`)
3. Commit your changes (`git commit -m 'Add some AmazingFeature'`)
4. Push to the branch (`git push origin feature/AmazingFeature`)
5. Open a Pull Request

📝 License

This project is licensed under the MIT License - see the LICENSE file for details.

🙏 Acknowledgments

- Thanks to the PandasAI team for the amazing library
- Google for providing the PaLM API
- Streamlit for the fantastic web framework

📞 Support

If you encounter any issues or have questions:

- 📧 Create an issue on GitHub
- 💬 Watch the YouTube tutorial for detailed guidance
- 🌐 Try the live demo first

Made with 💝 by Yash Kavaiya