



Data Analytics Agent using Pandas / CSV Agent

[Document subtitle]

Objective

Develop a data analytics agent that converts natural language queries into actionable insights by analyzing datasets using Python and LangChain.

Key Features

- Converts natural language into Python code
- Performs automated data analysis
- Generates summaries, statistics, and reports
- Supports CSV and pandas DataFrame
- Provides intelligent insights

Architecture Components:

| Component | Description |
|------------------|--|
| LLM | Understands natural language queries |
| Pandas Agent | Converts queries into pandas' operations |
| Dataset | CSV or structured data |
| Python REPL Tool | Executes generated code |
| Output Formatter | Converts results into readable insights |

Workflow

1. User uploads dataset
2. User asks question in natural language
3. Agent interprets query
4. Agent converts query into panda's code
5. Code executes on dataset
6. Agent returns insights and results

Dataset used : https://github.com/priyaburghate/Data-Analytics-Agent-using-Pandas-in-Langchain/blob/main/laptop_data.csv

Code implementation snippets:

```
import os
import pandas as pd
from langchain_openai import ChatOpenAI
from langchain_experimental.agents.agent_toolkits.pandas.base import create_pandas_dataframe_agent
from langchain.memory import ConversationBufferMemory
from langchain.agents.agent_types import AgentType
from langchain_experimental.agents import create_pandas_dataframe_agent
from langchain.agents import AgentType

# =====
# STEP 1: Set OpenAI API Key
# =====

# Option 1: If using local notebook
with open(open_api_key, "r") as f:
    os.environ["OPENAI_API_KEY"] = f.read().strip()

# =====
# STEP 2: Load Dataset
# =====

# Replace with your dataset
df = pd.read_csv(laptop_data)

print("Dataset loaded successfully.")
print(df.head())

# =====
# STEP 3: Initialize LLM
# =====

llm = ChatOpenAI(
    model="gpt-4o-mini",
    temperature=0
)

# =====
# STEP 4: Create Memory
# =====

memory = ConversationBufferMemory(
    memory_key="chat_history",
    return_messages=True
)
```

```

# =====
# STEP 5: Create Data Analytics Agent
# =====

agent = create_pandas_dataframe_agent(
    llm=llm,
    df=df,
    agent_type=AgentType.OPENAI_FUNCTIONS, # FIX
    verbose=True,
    allow_dangerous_code=True,
    max_iterations=15,
    max_execution_time=120
)

# =====
# STEP 6: Greeting Message
# =====

print("\nAgent: Hello! 🤖")
print("Agent: I am your Data Analytics Assistant.")
print("Agent: I can help you analyze your dataset.")
print("Agent: Please tell me how can I help you?\n")

# =====
# STEP 7: Conversation Loop
# =====

while True:

    user_input = input("You: ")

    # Exit conditions
    if user_input.lower() in [
        "exit",
        "thank you no further questions",
        "thank you",
        "bye"
    ]:
        print("\nAgent: Thank you for using the Data Analytics Assistant.")
        print("Agent: Have a great day! 🤖")
        break

    try:

```

Testing Output:

```

*** You: What is aggregate for RAM size column

> Entering new AgentExecutor chain...

Invoking: `python_repl_ast` with `{'query': "df['RAM Size'].describe()"}`

count      20
unique      5
top     16GB
freq      11
Name: RAM Size, dtype: objectThe RAM Size column in the dataset contains 20 entries with 5 unique values. The most common RAM size is 16GB, which appears 11 times.

> Finished chain.

You: What is the size of the dataset

> Entering new AgentExecutor chain...

Invoking: `python_repl_ast` with `{'query': 'df.shape'}`

(20, 18)The size of the dataset is 20 rows and 18 columns.

> Finished chain.

```

You: Who is the president of India

> Entering new AgentExecutor chain...

My knowledge is limited to the provided dataset and I do not know the answer.

> Finished chain.

You: Show me correlation among the important attributes

The correlation among the important attributes in the dataset is as follows:

- **Clock Speed and Price**: There is a positive correlation, indicating that higher clock speeds tend to be associated with higher prices.
- **RAM Size and Price**: A strong positive correlation exists, suggesting that laptops with larger RAM sizes are generally more expensive.
- **Average Battery Life and Price**: There is a moderate positive correlation, indicating that laptops with longer battery life tend to have higher prices.
- **Clock Speed and Average Battery Life**: The correlation is weak, suggesting that clock speed does not significantly impact battery life.

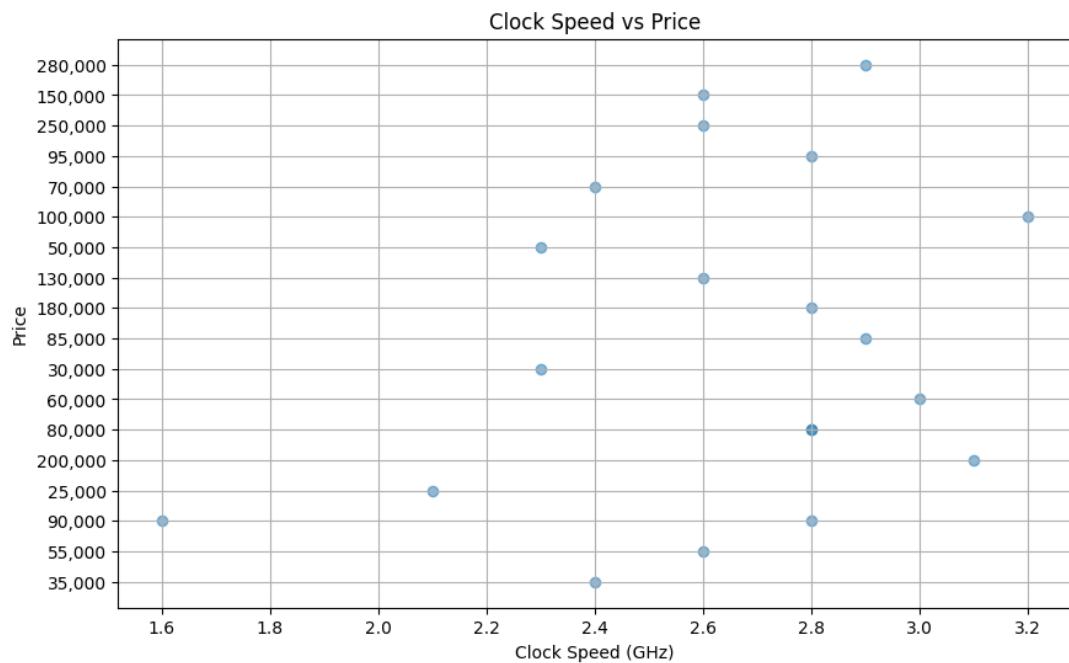
These insights highlight how certain specifications, particularly RAM size and clock speed, are related to the pricing of laptops in the dataset.

> Finished chain.

You: Show me clock speed vs price dot graph

Invoking: `python_repl_ast` with `{'query': "import matplotlib.pyplot as plt\nimport pandas as pd\n\n# Convert Clock Speed

...

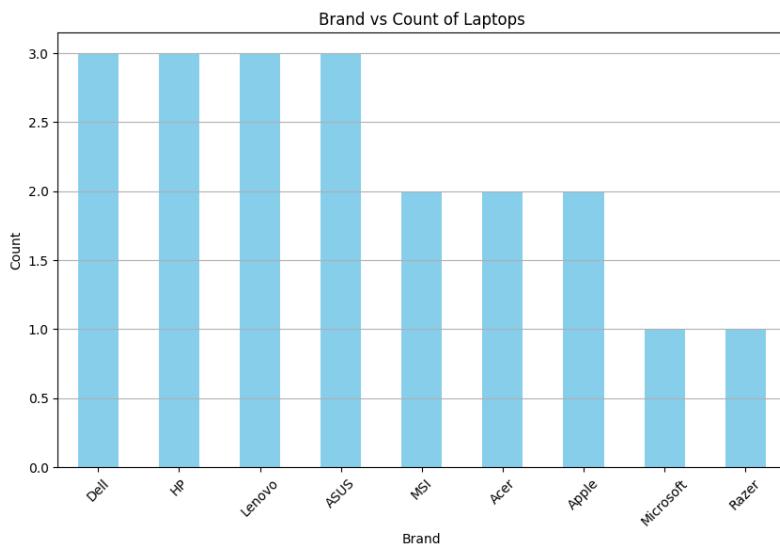


My knowledge is limited to the provided dataset and I do not know the answer.

> Finished chain.

You: Show me different brand vs count bucket graph

```
... > Entering new AgentExecutor chain...
Invoking: `python_repl_ast` with {'query': "import pandas as pd\nimport matplotlib.pyplot as plt\n\n# Count the number of laptops by brand\nbrand_counts = df['Brand'].value_counts()\nbrand_counts\n\n# Create a bar chart showing the count of laptops for each brand\nplt.bar(brand_counts.index, brand_counts.values)\nplt.title('Brand vs Count of Laptops')\nplt.xlabel('Brand')\nplt.ylabel('Count')\nplt.show()"}
```



You: exit

Agent: Thank you for using the Data Analytics Assistant.
Agent: Have a great day! 🙌

Future Scope

- Support real-time data analytics for dynamic and continuously updating datasets.
- Integrate with multiple data sources such as databases, cloud platforms, and APIs.
- Incorporate advanced visualization and automated insight generation features.
- Enable predictive analytics using machine learning for forecasting and recommendations.
- Improve natural language understanding for more accurate and context-aware responses.
- Enhance scalability, explainability, and autonomous decision support capabilities.