1. End-to-End Agent Workflow

```
flowchart LR
    A[Ingestion Agent] --> B[Profiling Agent]
    B --> C[EDA & Visualization Agent]
    C --> D[Cleaning Agent]
    D --> E[Feature Engineering Agent]
    E --> F[Model Training Agent]
    F --> G[HPO Agent]
    G --> H[Evaluation Agent]
    H --> I[Explainability Agent]
    I --> J[Reporting & Dashboard Agent]
    J --> K[Deployment & Monitoring Agent]
    K --> L[Feedback & Iteration Agent]
    L --> M[Knowledge Agent]
    style A fill:#f9f,stroke:#333,stroke-width:1px
    style C fill:#ff9,stroke:#333,stroke-width:1px
    style J fill:#9ff,stroke:#333,stroke-width:1px
```

---

2. Agent & Task Breakdown

Agent	Key Tasks	Tools & Services
1. Data Ingestion	• Connect to sources (CSV/SQL/Delta/ADLS)• Validate schema• Snapshot raw data	pandas, Azure Data Factory, Blob Storage
2. Data Profiling	• Compute column types, cardinalities• Detect high-null or constant columns• Sample value distributions	pandas-profiling, Azure ML DataPrep
3. EDA & Visualization	• Chart Recommendation (heatmaps, bar, line, scatter, box)• Generate static plots (matplotlib)	matplotlib, Seaborn via code interpreter
	• Produce interactive dashboards (Plotly/Dash or Power BI embeds)• Time-series decomposition plots	Plotly, Power BI, Streamlit

	• Automatic commentary: “Sales peaked in Q4…” via OpenAI	Azure OpenAI (GPT-4), Autogen prompt chains
4. Data Cleaning	• Missing value strategies• Outlier detection & treatment• Type conversions	pandas, Autogen decision trees
5. Feature Engineering	• One-hot / target encoding• Scaling / normalization• Text embeddings (TF-IDF, BERT)• Date/time features	scikit-learn, Azure ML Feature Store
6. Model Training	• Baseline models (linreg, tree)• AutoML runs for quick baselines	Azure ML, scikit-learn
7. Hyperparameter Tuning	• Grid/Bayesian search via HyperDrive• Parallel experiment orchestration	Azure ML HyperDrive, custom Python
8. Model Evaluation	• Metrics (RMSE, AUC, F1)• Confusion matrices & ROC curves• Statistical tests (t-tests, KS-tests)	scikit-learn, matplotlib for ROC/confusion plots
9. Explainability	• SHAP/LIME value computation• Global vs. local explanations• Feature importance plots	SHAP, LIME, code interpreter + OpenAI for summaries
10. Reporting & Dashboard	• Assemble narrative report (Markdown/HTML/PDF)• Embed key static + interactive visuals• Publish to Power BI	Azure Functions, Power BI REST API, Jinja2
11. Deployment & Monitoring	• Containerize model (Docker/ACI)• Expose REST endpoint• Set up telemetry & drift alerts	Azure Container Instances, App Insights, Monitor
12. Feedback & Iteration	• Collect user feedback on dashboards• Trigger retraining or feature tweaks	Azure Forms, Logic Apps, Autogen loops
13. Knowledge & Memory	• Store artifacts & metadata• Answer retrospective queries (“last best model…”)	Azure Cognitive Search, Cosmos DB

---

3. Visualization-Centric Agents

1. **Chart Recommendation Agent**

   - **Input:** DataFrame schema + sample rows

   - **Output:** List of suggested chart types (`“histogram”`, `"heatmap"`, `"scatter”`) with rationale

    - **How:** Prompt GPT‑4 via Autogen with examples of chart-purpose mapping.

2. **Static Plot Agent**
    - **Input:** Selected chart spec + data
    - **Output:** Matplotlib/Seaborn PNGs
    - **How:** Execute code in a sandboxed interpreter, return image URLs.
3. **Interactive Dashboard Agent**
    - **Input:** Set of visuals + data endpoints
    - **Output:** A live Streamlit/Dash app or embedded Power BI report link
    - **How:** Use Azure Functions to deploy the dashboard and publish URL.
4. **Insight Extraction Agent**
    - **Input:** Visual outputs + metrics
    - **Output:** Natural‑language insights (“Correlation between X and Y is strong…”)
    - **How:** Use Azure OpenAI to parse chart statistics and draft summaries.

---

4. Orchestration Patterns

- **Sequential Pipeline** for full runs (ingest → profile → EDA → … → deploy).
- **On‑Demand Sub‑Pipelines**: Analysts can call just “EDA & Visualization” or “Explainability” for new data.
- **Event‑Driven**: Feedback Agent triggers a retraining cycle when drift crosses threshold.
- **Parallel Swarms**: Multiple Feature Engineering agents generate candidates concurrently; Orchestrator ranks them.

---

5. Next Steps

1. **Define Agent Contracts**
    - For each agent, specify JSON schemas: inputs, outputs, tool calls.
2. **Kick off MVP**
    - Build Ingestion → Chart Recommendation → Static Plot → Insight Extraction chain.
3. **Iterate**
    - Add interactive dashboards and downstream agents one at a time.
4. **Test & Validate**
    - Use a sample dataset (e.g., Titanic, NYC taxi) end-to-end.
5. **Submit**
    - Fill out your `project.yml`, link code, screenshots of each agent in action, and a short demo GIF.

This design ensures **every** Data Science task—**from raw data profiling through advanced visualization, model building, interpretability, deployment, and feedback loops—is handled by a dedicated, testable agent**.