# **📌 AI-Powered Data Analysis Project – Full Summary & Details**

**Contents**

[**📌 AI-Powered Data Analysis Project – Full Summary & Details 1**](#_Toc191731992)

[🔹 Project Overview 3](#_Toc191731993)

[🔹 Project Workflow 3](#_Toc191731994)

[🔹 Tools & Technologies Used 4](#_Toc191731995)

[🔹 Final Outcome 5](#_Toc191731996)

[🔹 Future Improvements 5](#_Toc191731997)

[🔹 Conclusion 5](#_Toc191731998)

**🔹 Project Overview**

The **AI-Powered Data Analysis Project** is an end-to-end machine learning pipeline that automates **data preprocessing, exploratory data analysis (EDA), model training, evaluation, and API deployment**. The project is structured using **Python, GitHub Actions CI/CD, and FastAPI** to enable efficient data analysis and machine learning model deployment.

**🔹 Project Workflow**

The project follows a structured workflow:

1️⃣ **Project Setup & Repository Creation**

* A **GitHub repository** was initialized for version control.
* Required directories: scripts, data, models, app were created.

2️⃣ **Virtual Environment & Dependencies Setup**

* A **Python virtual environment** was created (venv).
* Dependencies were listed in requirements.txt and installed.

3️⃣ **Data Preprocessing & Cleaning**

* **Handling missing values**, encoding categorical variables, and normalizing data.
* scripts/data\_preprocessing.py reads raw data (sample\_data.csv), processes it, and saves processed\_data.csv.

4️⃣ **Exploratory Data Analysis (EDA)**

* scripts/eda.py performs **statistical analysis** and **visualizations**.
* Generates plots using **Matplotlib & Seaborn**.

5️⃣ **Machine Learning Model Training**

* scripts/train\_model.py trains a **Random Forest classifier**.
* Other models (SVM, Neural Network) were included as options.
* The trained model is saved in models/random\_forest.joblib.

6️⃣ **Model Evaluation**

* scripts/evaluate\_model.py loads the trained model and evaluates it using **classification metrics**.

7️⃣ **API Deployment using FastAPI**

* app/main.py sets up an API endpoint (/predict) for making predictions.
* **Uvicorn** is used as the ASGI server for running the API.

8️⃣ **GitHub Actions CI/CD Integration**

* A workflow (.github/workflows/python-package.yml) was set up for **automated testing and deployment**.
* The pipeline includes:
  + **Checkout the latest code**
  + **Set up Python environment**
  + **Install dependencies**
  + **Run data preprocessing**
  + **Train the model**
  + **Deploy API tests (optional)**

9️⃣ **Troubleshooting & Fixes**

* **Merge conflicts were resolved**.
* **GitHub Actions failures due to missing checkout steps were fixed**.
* **Syntax errors in YAML workflow were corrected**.

🔟 **Final Confirmation**

* The GitHub Actions workflow now runs successfully.
* The trained model is available for real-time predictions via FastAPI.

**🔹 Tools & Technologies Used**

| **Category** | **Tools/Technologies Used** |
| --- | --- |
| Programming Language | Python |
| Libraries | NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn, Joblib, FastAPI, Uvicorn |
| Machine Learning Models | Random Forest, SVM, Neural Network |
| CI/CD & Automation | GitHub Actions |
| Version Control | Git & GitHub |
| API Deployment | FastAPI & Uvicorn |
| Data Storage | CSV files (processed data, sample data) |

**🔹 Final Outcome**

✔ **Successfully set up a GitHub repository with structured ML scripts**  
✔ **Implemented an AI-powered pipeline for data preprocessing, model training, and API deployment**  
✔ **Automated GitHub Actions workflow for CI/CD testing and deployment**  
✔ **Trained a machine learning model to predict outputs based on input features**  
✔ **Deployed an API endpoint for real-time predictions**

**🔹 Future Improvements**

🚀 **Deploy API on AWS, Heroku, or Render for public access**  
🚀 **Enhance CI/CD with unit testing for API requests**  
🚀 **Improve model selection with hyperparameter tuning**  
🚀 **Optimize data preprocessing for larger datasets**

**🔹 Conclusion**

This project showcases the **full ML workflow**, from **data preprocessing to model deployment**, with **automated testing** using **GitHub Actions CI/CD**. It demonstrates the use of **FastAPI for real-time predictions**, ensuring a **scalable and reproducible** AI solution.