

This document will give you comprehensive about the files uploaded in github as a part of submission. It also covers proposed solution.

### **Step 1:**

- IDE Used: Visual studio and Jupyter notebook inside VS Code
- Create new venv and activate it using python or conda
- Using pip, you can install requirements.txt file (using command: pip install -r requirements.txt)

### **Step 2: Data Sample files and Data Generation Approach**

- Inside your workspace – create two folders:
  - csv\_files
  - json\_files
- Paste all csv\_files and json\_files inside their respective folders.
- Note: Change the path as per your need in order to open/analyze files
- Json data files:
  - sample\_diverse\_dataset.json
    - Contains diverse herd information (yak name, age, health and behavior)
  - sample\_json.json
    - Contains herd information (yak name, sex, age)
  - sample\_order\_data.json
    - Contains order information for 100 random customers
    - It shows customer name, order, date
  - sample\_stock\_data.json
    - Contains 100 stock samples (milk, skins)
  - customer\_order\_fullfillment\_results.json
    - Contains order fulfillment status for 100 customers from sample\_order\_data.json with corresponding orders in sample\_stock\_data.json
  - Sample\_diverse\_dataset\_updated.json
    - More features added for behavior analysis
  - sample\_diverse\_dataset\_recommendation.json
    - Comprehensive data for recommendation analysis
- **CSVs data files:**
  - sample\_diverse\_dataset.csv – for querying using NLP agent
  - sample\_json.csv – for querying using NLP agent
  - sample\_order\_data.csv – for querying using NLP agent
  - sample\_stock\_data.csv – for quering using NLP agent
- **Data Creation Approach:**
  - REFERENCE FILE: data\_generator.ipynb
  - Based on the sample data provided in the tasks, 100 samples are created randomly for each json.

- These json files are further utilized for every other tasks – core\_functionality, AI models, Behavior analysis
- CSV files are just for NLP Query Agents

## Other Python Files

- **core\_functionality\_solution.ipynb**
  - this file solves core functionality needed for the tasks which includes:
    - Data Preprocessing
    - Stock and Herd Management Functions
    - Order Fulfillment Logic
  - Note: code logics can be referred via comments
- **anomaly\_detection.ipynb**
  - ML model for anomaly detection
    - Model Used – Unsupervised Learning ML Model: IsolationForest
    - Reason to choose this model:
      - Effectiveness in Handling Outliers
      - Robustness to Noise and Irregularities
      - Efficient Computation
      - Parameter-Free Approach
      - Handling High-Dimensional Data
      - No Assumptions about Data Distribution
      - Effective in Unsupervised Learning Scenarios
  - **Note:** Testing has been done using inference data
  - **Note:** Model has been evaluated on accuracy, false positive rates
- **behavior\_analysis.ipynb**
  - ML model for behavior analysis
  - **Note:**
    - The provided behavior analysis model attempts to predict yak behavior based on 'Age' and 'Health' attributes. While it's a step toward understanding yak behavior, fulfilling the statement to predict and comprehend their behavior over time requires a more comprehensive approach and more features. These features were not present as a part of the sample data
    - Additional relevant features, such as environment, diet, social interactions, or seasonal changes, might provide more comprehensive insights.
  - Model used: Binary Classification model where yak behavior is analyzed with age and health attributes
  - **Note:** Testing has been done on selecting random data samples from test data and check the model predictions (ground\_truth\_behavior vs model\_predicted\_behavior)
- **behavior\_analysis\_updated.ipynb**

- New experiments with randomly generated new features to make behavior analysis more robust
- recommendation\_analysis.ipynb
  - recommendation analysis of yak health based on observed attributes
- final\_app\_agent\_nlp.py
  - In the terminal, type streamlit run final\_app\_agent\_nlp.py
  - **Note:** Make sure to use your own OPENAI API KEY from OPENAI
  - a webpage has been created to query different data related csv files
  - Functionality:
    - You can download multiple CSVs at once
    - You can choose on what csv you need to perform query. Accordingly agent will provide you the answers
    - Agents' modules are used instead of chains modules of LangChain. Agents are not rule based models unlike chains where users have to define a set of prompts in order to get answers from their query. Surprisingly, agents handles this straightaway
    - **Note:** You can query any questions from any csv files
    - **Note:** You cannot update any value. I doubt if this functionality exists or not
    - **Note:**
      - My OpenAI free credits are finished, so I am not getting responses back from LLM. Let me fulfill it, I will update it if a demo needs to be shown
      - However, check simple and cool website design to upload and view csvs

Condensed steps for deploying the Yak Shop with AI/ML features on GCP:

## Deployment Steps

### 1. Data Processing:

#### Google Cloud Storage:

- Upload the JSON file containing herd information to Google Cloud Storage.

#### Compute Engine:

- Create a Compute Engine instance to run the data processing program.
- Install necessary dependencies and libraries for Python or any preferred programming language.
- Write a script that reads the JSON file path and the elapsed time parameter.

- Use GCP SDKs or libraries to interact with Google Cloud Storage to access the JSON file
- Process the data to simulate changes in the herd after the specified time.

## 2. Stock and Herd Management:

### Google Cloud Functions or App Engine:

- Develop APIs or web services for stock and herd management.
- Utilize Google Cloud Functions or App Engine to host these APIs.
- Connect these services to the processed data obtained from Data Processing step.
- Implement endpoints to calculate and display milk and skin stock after T days.
- Create endpoints to view the herd after T days, including yak details.

## 3. Order Fulfillment:

### Google Cloud Firestore or Cloud SQL:

- Set up a database to manage orders, customer details, and stock availability.
- Create tables/collections to store order details and available stock.
- Develop APIs or services using Cloud Functions or App Engine to handle order requests.
- Implement logic to check stock availability based on incoming orders.
- Return appropriate HTTP status codes and order details based on stock availability.

## 4. AI/ML Anomaly and Behavior Analysis:

### Google Cloud AI Platform:

- Train anomaly detection and behavior analysis models using Google Cloud AI Platform's machine learning services.
- Prepare and preprocess data for training these models.
- Deploy trained models as endpoints on the AI Platform.
- Integrate these endpoints with your Yak Shop application to monitor yak health and behavior.
- Retrieve predictions or analysis results for anomalies and behavioral insights.

**Note\_1:** Throughout these steps, ensure proper authentication, access control, and permissions are set up using GCP IAM (Identity and Access Management) to secure access to resources and services.

**Note\_2:** configure networking and API endpoints appropriately to enable communication between different components of the Yak Shop application on GCP.

**Note\_3:** Always monitor and test the deployed functionalities to ensure they perform as expected and consider using GCP's monitoring and logging services to track application behavior and performance.