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_1.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 downloaded 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

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

```
from google.cloud import storage

# Create a storage client
storage_client = storage.Client.from_service_account_json('path/to/your/service_account_key.json')

# Get the bucket containing your JSON file
bucket = storage_client.get_bucket('your_bucket')
```

```
# Get the JSON file from the bucket
blob = bucket.blob('your_file.json')

# Download the file contents
blob.download_to_filename('local_file.json')

# Read the contents of the downloaded file
with open('local_file.json', 'r') as file:
    json_data = file.read()

# Use the JSON data as needed
print(json_data)
```

- 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.

Scope of Improvement:

Make Yak Web shop Scalable:

- **Database Management:** Employing scalable databases or data partitioning techniques. This can help us to handle increased product inventory, customer data, and transactions.
- **Cloud Infrastructure:** Utilizing cloud services (AWS, GCP, Azure) for scalability in web hosting, storage, and computation resources based on traffic fluctuations.
- **Elasticity in Design:** Ensuring architecture which allows for easy scaling of the system components in order accommodate increased user traffic and product listings

AI Model Selection for Large Datasets in Yak Web shop:

1. **Anomaly Detection**
    a. **Distributed Computing:** Implement distributed frameworks like Apache Spark to handle scale dataset parallel and efficiently for anomaly detection
    b. **Stream Processing:** Utilize stream processing architectures (e.g. Apache Kafka) to process continuous data streams in real – time, enabling rapid anomaly detection as new data arrives. This can be used when we have real time data.
    c. **Feature Selection:** Optimize anomaly detection algorithms by employing feature selection techniques like PCA, variance threshold, correlation threshold to focus on relevant features and reduce computation overhead
    d. **Other Models:**
        i. One class SVM
        ii. Density Based Methods (Identify clusters and anomalies in large data set):
            1. K means Clustering,

2. DBSCAN

iii. Deep Learning (Require more computation resources)

1. VAEs,
2. RNNs for anomaly detection

2. Behavior Analysis

a. Assuming Unsupervised Learning: (K Means, DBSCAN)

b. Sequence Analysis Models (RNNs)

i. Useful t to capture sequential behavioral data over time

ii. Can capture temporal dependencies in large-scale datasets

3. Recommendation System:

a. Matrix Factorization:

i. Traditionally used for collaborative filtering in recommendation system but can be applied to herd management as well.

ii. When applied on yak health records or behavior patterns, to extract latent factors contributing to herd health and behavior changes, enabling recommendations for appropriate health interventions or adjustments in feeding schedules

b. Content Based Filtering:

i. Basically recommends items based on item features and user preferences. Utilizing content – based analysis to recommend specific feed types or health interventions based on specific yak features (age, health history, environmental conditions, etc). This aids in suggesting personalized interventions or adjustments in the feeding schedules

c. Deep Learning Models (Neural Collaborative Filtering)

i. It can capture complex relationships within the herd data

ii. Apply these architectures to comprehend relationships between various herd health parameters, environmental factors, and interventions.

iii. This enables the identification of nuanced patterns that affect herd well-being, aiding in suggesting precise feeding schedules or targeted health interventions for optimal herd management.