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
* **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)
* 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
    - **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