Here are the instructions as discussed. I wrote down these instruction quickly and let me know if anything is unclear.

Please use markdown to document your notebook and indicate sections clearly.

Hints:

* It may be necessary to correctly timestamp data so that Beam is able to reason about it in the stream.
* Consider using a Python with pandas dataframes to test and verify your approaches.
* part 5.1, an initial pipeline is expected to produce two file outputs - you cannot do this without performing a specific operation.
* part 6.3 It is advisable to choose a simple modeling approach.

# Part 2: Streaming Applications

**4: Apache Beam Analytics** Consider the data provided in the streaming (namely [*users.csv*](https://storage.googleapis.com/bdt-beam/users_v.csv)) alongside *orders* data (namely [*orders.csv*](https://storage.googleapis.com/bdt-beam/orders_v_2022.csv)). Combine these 2 input files in order to provide insights into the purchasing behaviours of the retailer’s customers.

1. Join the input files as streams using CoGroupByKey.
2. Perform a transformation that determines the average number of orders for female and male customers, respectively. Output the result as text in the notebook.
3. Duplicate the code of previous question (in a new cell) and split your pipe to produce/emit the total number or orders processed as an additional text output (i.e. output the number of orders processed as well the average orders for female and male customers).
4. Generate a graph (using the Beam library) of your pipeline
5. Perform a transformation that groups users into age groups [16-26), [26-36), [36-46), [46-56), and determine the total number of orders placed by customers in each age group.
6. Determine the total number of times that spinach was purchased within the [16-26), [26-36), [36-46), [46-56) age groups.

## 5: Stream Analytics Cont.

1. Write a pipeline which calculates the average number of orders over seven and thirty day rolling windows (where the *period* of each window is one day).
2. Modify this pipeline to write the output files to Parquet instead of CSV.

**6: Anomaly Detection on Streams** Consider the [available data](https://storage.googleapis.com/bdt-beam/uni-anom-line.csv) containing the measure- ment of a simple metric over time. Although we will be considering this problem in a bounded setting, in reality our solution is required to run on a continuous stream of data.

1. Produce an *exploratory data analysis* (EDA) notebook to investigate the initial data. Summarise your conclusions/insights in the topmost cell block (in markdown).
2. Select a simple Scikit-learn method for detection and motivate the choice.
3. Create a notebook demonstrating the fitting of the model (if applicable), as well as the test/inference step on a stream using the Sklearn RunInference transform.
4. Evaluate the success/correctness of your implementation given the ground truth.