Final problem:

Problem statement:
The goal of the problem is to generate innovative insights from order dataset and also to increase the average order value (AOV). The dataset for this competition is a flat-file describing customers orders over time. The dataset is anonymized and contains a sample of over 6 million grocery orders from more than 30000 Milkbasket users. The only information provided about users is their sequence of orders and the products in those orders. All of the IDs in the dataset are entirely randomized, and cannot be linked back to any other ID
For each user, we provide more than 300 of their orders, with the sequence of products purchased in each order. We also provide the time of the day the product was added to the basket and the order was assumed to be created at that time itself.

File descriptions:
Full data set: https://blume-hackathon.s3.ap-south-1.amazonaws.com/milkbasket_hackathon_data.zip
Sample file: https://blume-hackathon.s3.ap-south-1.amazonaws.com/milkbasket_hackathon_sample_data.csv

customer_id,manufacturer_id,society_id,city_id,route_id,store_id,order_id,order_date,category_id,subcategory_id,product_id,product_quantity,selling_price_per_unit,total_cost,subscription,product_addedtobasket_on
2698080,1122016,1127168,1120112,1121456,1120112,338048928,2018-05-01,1122576,1125264,1120336,4,16.01,64.04,0,2018-04-30 16:23:46
1134224,1134336,1120224,1120112,1120336,1120112,337533168,2018-05-01,1122576,1125152,1939280,1,8.79,8.79,1,2018-04-19 01:30:02
3686704,1150128,1126160,1120112,1123472,1120112,338235520,2018-05-01,1123472,1130640,1681456,1,25.0,25.0,0,2018-04-30 22:26:07

description of columns:
customer_id = id of the user/customer
manufacturer_id = brand manufacturer id
society_id = id of society where the order was created
city_id = id of city from where the product was ordered
route_id = id of route which the delivery took
store_id = if of store from where the product was ordered
order_id = id of order
order_date = date of order
category_id = category id (e.g. beverages)
subcategory_id = subcategory id (e.g. milk)
product_id = id of product
product_quantity = total quantity of product added
selling_price_per_unit = selling price of product (after discounts)
total_cost = total cost (product_quantity*selling_price_per_unit)
subscription = boolean flag to denote if the product was a subscribed product (i.e. recurring)
product_addedtobasket_on = time when the product was added to customers basket

Bonus: Create a mobile app that is powered on the above order data for a customer and place suggestions to the customer to buy similar or related products.

Evaluation: Submissions will be evaluated based on innovation, logic, the breath of the approach to the problem, teamwork.
Tech stack: Python, Numpy, Pandas, Scikit learn, Keras, Apache spark
Ideal team: 4 members with Data crunching skills, Stats, Programming in python/java, APIs, serverless