Table of Contents

Feedback | Group 2

Milestone 1 | 2Oct-13Oct

1. **Define the problem:** done
   - Overall Good
   - By the end of 2nd milestone, it is expected to specify the CLV calculation method (we are going to cover it)
   - There is a need to define business settings
2. **Finalizing roles:** done
3. **Create a product roadmap and prioritize functionality (items):** not done
   - The Front-End part is confusing. Are going to have a UI Layout? Where is DB developer?
   - In must-have, you have mentioned Visualizations?
4. **Creating the GitHub repository included readme.md and `.gitignore` (for Python) files:** done
   - done partially: during the remote repository initialization, you should have selected add .gitingorne with the Python option
5. **Create a virtual environment in the above repo and generate requirements.txt (`venv` must be ignored in git)** done
   - it seems you did with conda create, without adding `--no-default-packages` option. As a result, we have a bunch of extra packages.
   - please fix it and push it to GitHub by the end of Milestone 2
6. **Push *point 1, point 3, point 5 (requirements.txt)*.** not done:
   - see point 5
7. **Complete the first chapter of Developing Python Packages:** done
   - completed by everyone
8. **Create a private Slack channel in our Workspace and name it Group-{number}** done
9. **Schedule a call with me and Garo or come during the office hours:** done

By the end of the Milestone 2, you must complete the tasks mentioned above. Feel free to reach out if you have any questions.

- CLV calculation method
- Business Model
- Add high-level tasks for DB developer (this one you will find on Milestone 2 as well)
- Fix requirements.txt

Grade: 5/10

Milestone 2 | 16Oct-27Oct

Fixes From the Milestone 1

I can see that you have managed to fix:

- The requirements.txt
- `gitignore`

Milestone 2

1. **DB developer:**
    - Design the database using Star schema (provide ERD): done
    - Insert Sample to data done
2. **Data Scientist:**
    - Complete data generation/acquisition/research: done
    - Select data from DB: done
    - Insert data to DB: done
    - I couldn't reproduce data insertion as csv files are in data_csv folder,unlike in your code
3. **API developer:**
    - Select data from DB done
    - Insert data to DB done
    - Update data in DB wrong arguments
4. Finish the second chapter of Datacamp course done
5. Finalize file/folder structure: relative imports must work properly not done
    - docs folder: putting all the documents there not done
    - models folder: putting modeling-related classes, functions not done
    - api folder: api related stuff not done
    - db folder: db related stuff not done
    - initialize `__init__.py` files accordingly (see Datacamp assignment chapter 1 and chapter 2) not done
    - logger folder: I will provide this module done

I can see only Anahit's contribution on GitHub

In order to improve you performance I would recommend:

- approach the datacamp course seriously (it is obvious You are just taking the hints and completing it)
- come to office hours
- request calls **in advance**
- if you are stacking on one problem too long, it simply means you are doing it wrong: the goal of the project is not a panishment
- no need to have too much code, without proper environment setup

By the end of the 3rd Milestone you must fix folders and their relationships.

If you manage the complete the above points by Friday(before the class) you will get **20/20**

Grade: 10/20

Milestone 3 | 30 Oct-10 Nov

1. Complete things from *Milestone 2* done
2. Remove M2 M1 folders, we need to have one folder- the name of the package, and its subfolder-modules done
3. Finish the **third** chapter of Datacamp course (please complete only the 3rd one) done by everyone
4. **API Developer:** done
    - Create a `run.py` file for an API (find the minimum workable example here)
    - Test it on swagger
    - following request types must be available to test (GET, POST, PUT), will provide more details on Friday.
5. **DB developer:**
    - complete/fix the methods from `SQLHandler()` class done
    - finalize the documentation for `schema.py` by using `pyment` package done
    - finalize the documentation for `SQLHandler()` by using `pyment` package done
6. **Data Scientist:** start working on modeling part, by selecting the date from SQL DB done
    - we just need to run sample model and store the output to sql

You have done really good job guys!

Grade: 30/30

1. Complete the **final (4th)** chapter of Datacampe course
2. Combine the Modeling notebook into a Python class and move to the `CLV_Analysis/models` folder. Make sure to import the model and model-related stuff in the `CLTV.ipynb` file. `from CLV_Analysis.Models import YourModel`
3. **Product manager:** Design 2 use cases for the package and communicate with the team:
    - Write down it in `readme.md`
    - keep only the the relevant points
    - In the `readme.md` provide the steps need to be done in order to use the package
4. **Other members:** If needed restructure your parts in order to meet the above requirements
5. Finalise the documentation for each `class/module/method/function`

Optional for the Milestone 4 , mandatory for the group project:

1. Create an `example.ipynb` file (out of the package) and implement all the package's functionality (make sure to make it chunk by chunk to convert it `reveal.js presentation). This is going to be part of the demo.
2. As soon as you finish the documentation us `Mkdocs` in order to generate docs.html file, which is going to be hosted on GitHub
3. Publish your package to `pypi.org`

Final Feedback

Group Project Scope

- Finding a Marketing related problem
- Understanding the methodology of the analysis
- Building a Python package with following mandatory modules:
    - Predictive Model (component)
    - DB
    - API
    - Logging (provided by me)
- Post to `Pypi.org`

Submission format

In the Github Repository, the following structure must be available

```
| GitHubRepo
    | Docs
    | PackageName
        |SubPackage_1
            modlule1
            __init__.py
        |SubPackage_2
            module2.py
            __init__.py
        __init__.py
        utils.py
    other files (.gitignore, *config files)
    readme.md
    requirements.txt
    setup.py
    example.py/ipynb (Demonstrate all the funcionality)
```

Submision format is correct.

Grading Methodology

Group Project is going to be graded according to the following poins:

1. **Topic Relevancy:** matched but not correctly demonstrated
2. **Team Work:** I can see the contrubtutions from each member
3. **Availability of Documentation:** Perfect
    - Description of each `function()/method()`:
        - Parameters: description/docstrings
        - Returns: what do you expect as a return?

    - Description of Classes:
        - Use *dunder methods:* `__repr__`, `__str__`, for nice Class formulation
        - Describe the class
    - converting into a webapp using `mkdocs` or any alternative
4. **The code must run without any errors:** OK
    - logical
    - syntax
    - runtime error
5. **The availability of a Predictive Element** CLV
6. **Endpoints solving/touching the business problem** OK
7. **Successfully hosted on** `Pypi` Done

Final Feedback

Technically you have done everything which was required. However you could have done way better by simply trying to understand what do you need to do at first.

It is obvious that there was a communication problem between **product/project manager** and the **team**. There is a gap between business/marketing and package logic.

Grade

- **Grade from the Milestones:** 90
- **Grade from the Presentation:** 300/300
- **Final Grade: 390**