Table of Contents

Feedback | Group 4

Milestone 1 | 2Oct-13Oct

FEEDBACK FOR MILESTONE 1 | Group 4

1. **Define the problem: ** done
    - Well defined!
2. **Finalizing roles:** done
3. **Create a product roadmap and prioritize functionality (items)** done
    - I really liked the roadmap and the set of tasks you are going to receive from me almost the same as in your roadmap
4. **Creating the GitHub repository included readme.md and .gitignore (for Python) files:** done
5. Create a virtual environment in the above repo and generate requirements.txt (venv must be ignored in git) done
6. Push *point 1, point 3, point 5 (requirements.txt).done
7. Complete the first chapter of Developing Python Packages done
    - Completed by everyone (except Vahagn Tovmasyan (-3 points))
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 office hours. done

Continue, according to the roadmap and also add the tasks for milestone 2 required by me,

Grade: 10/10 Good job!

Milestone 2 | 16Oct-27Oct

Fixes From the Milestone 1

Fixes were note required!

Milestone 2

1. **DB developer:**
    - Design the database using Star schema (provide ERD): done
    - Insert Sample to data done
    - the structure is wrong
2. **Data Scientist:**
    - Complete data generation/acquisition/research: done
    - Select data from DB: done
    - Insert data to DB: done
3. **API developer:**
    - Select data from DB not done
    - Insert data to DB not done
    - Update data in DB not done
    - api module must be inside of the package; I cannot see your package stracture(just change the name etl to a relevant name). See, the etl forder is structured as pacakge, meaning that you should move api folder there with corresponding __ init __.pys
4. Finish the second chapter of Datacamp course done by everyone
5. Finalize file/folder structure: relative imports must work properly not done see point 5
    - 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 multiple contributors

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)
- start to work on group project before the deadline

Remember you are building a package, like in the Datacamp you must have following file structure:

```
| GitHubRepo
    | PackageName
        |SubPackage_1
            modlule1
```

```
            __init__.py
        |SubPackage_2
            module2.py
            __init__.py
        __init__.py
    setup.py
    example.py/ipynb (from PackageName import SomeModule)
```

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: 15/20

Milestone 2 | Revision

You have managed to fix all the requirments on time.

The things to fix though:

- `FillTables.py` must be out of the package (see above structure)
- Just change the name of the packaage and instead of `how_to_use.txt` update `readme.md` file.

Revised Grade: 20/20

Milestone 3 | 30Oct-10Nov

1. Complete things from *Milestone 2* done
2. Finish the **third** chapter of Datacamp course (please complete only the 3rd one) done by everyone
3. **API Developer:**
    - Create a `run.py` file for an API (find the minimum workable example here) done
    - Test it on swagger done
    - following request types must be available to test (GET, POST, PUT), will provide more details on Friday. done
4. **DB developer:**
    - complete/fix the methods from `SQLHandler()` class done
    - finalize the documentation for `schema.py` by using `pyment` package not done
    - finalize the documentation for `SQLHandler()` by using `pyment` package not done
5. **Data Scientist:** start working on modeling part, by selecting the data from SQL DB done
    - we just need to run sample model and store the output to sql
    - isstead of `table_names = ['State', 'PlanDetails', 'DayUsage', 'EveUsage', 'NightUsage', 'IntlUsage', 'CustomerMetrics']`, provide it a an argument, in order to make it dynamic
6. With **product manager** Work on application scenarios and come up at least two use cases. done

Grade: 25/30

1. Complete the missing parts from Milestone 3
2. Try to make it as general as possible
3. provide an example notebook file (`example.ipynb`) which will do:
    - data prep and ingestion with short explanation
    - modeling part with short explanation
    - running the api code
    - try to use short chunks in order to convert it later `reaveal js` presentation for the demo.
4. finish documentation for all
5. use `MkDocs` to generate html documentation (optional for this milestone, mandatory for group project)

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** Churn
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.

I

Grade

- **Grade from the Milestones:** 95
- **Grade from the Presentation:** 300/300
- **Final Grade: 395**