Feedback | Group 2

Table of Contents

Milestone 1 | Tasks

Problem Definition | 20 points

The problem is not described properly. Here are described the steps of **defining the problem** and **proposing a solution.**

- Broad Area of Interest
- Preliminary Research
  - Current trends
  - Opportunities
- Solution with Methodology
  - Data Collection
  - Analytical Techniques
  - Implementation Plan
- Expected Outcomes
- Evaluation Metrics

Grade: 5

Roadmap | 10 points

The roadmap seems realistic.

Grade: 10

Administrative Tasks | 5 points

- Roles are assigned
- A preliminary discussion with me was done
- Slack channel is created
- Github Repo is created

Grade: 5

Technical Tasks | 5 points

- Proper `.gitignore` file is available
- The requirements.txt file is available, indicating that `venv` was created
- The first chapter of the Package Development course is done by **everyone**

Grade: 5

Grade

Final Grade: 25/40

Milestone 2 | Tasks

Fix the problem statement from the first milestone.

Product and Project Manager | 40 points

1. Name your Python package: register to pypi
2. Install `mkdocs` package to start with the documentation
3. Database schema: Provide your product database structure (ERD)
4. Transform your project file structure according to the below tree

```
PythonPackageProject/ #githhub repo
├── yourpackagename/
│   ├── __init__.py
│   └── submodule1/ #database related
│       ├── __init__.py
│       └── submodule1_1.py
│   └── submodule2/ #model related
│       ├── __init__.py
│       └── submodule1_2.py
|     └── submodule3/ # api related
│       ├── __init__.py
│       └── submodule1_2.py
├── tests/
│   ├── __init__.py
│   ├── test_module1.py
│   └── test_module2.py
|── example.ipynb # showing how it works
|── run.py # in order to run an API
|── docs/ #this folder we need for documentation
|── .gitignore
|── requirments.txt
├── README.md
├── LICENSE
└── setup.py
```

Data Scientist and Data Analyst | 20 points

1. Simulate the data if you need
2. Try to use the CRUD functionality done by DB Developer
3. Work on modeling part using simple models

```
from yourpackage.submodule2 import modelname
```

Database Developer | 30 points

1. Create a DB and respective tables suggested by the Product Manager
2. Connect to SQL with Python
3. Push data from flat files to DB
4. Test the code provided here and complete the missing components
5. Add extra `methods` that you might need throughout the project:
    1. Communicate with PM and API Developer for custom functionality

```
from yourpackage.submodule1 import sqlinteractions
```

API Developer | 30 points

1. Communicate with DB Developer and PM in order to design the API
2. You can create dummy endpoints in the beginning, then communicate with PM as well
3. The following endpoints must be available:
    1. GET
    2. POST
    3. UPDATE

Check out this this repo.

```
from yourpackage.submodule2 import api
```

Milestone 2 | Feedback

Tasks from Milestone 1

You have updated the problem definition part. I would recommend making it a bit readable by using bullet points and text highlighters as well. Anyway, It is going to be part of the final presentation.

DataCamp

Done by everyone except Davit Khalatyan (-10 points for Davit)

Product and Project Manager | 40 points

1. I couln't find the `pypi` link. Put it in the README file
2. Couldn't find `mkdocs` package in the requirments.txt
3. Done
4. Partially Done
   - package file structure is correct
   - package usage must be one level higher in the GitHub repo(`marketing_group_project`). with this structure you would push your tests to `pypi`
   - The data folder must be under the GitHub repo not inside the package
   - `run.py` is missing
   - `docs` folder is missing

Grade: 30/40

Data Scientist and Data Analyst | 20 points

- The data was successfully simulated
- CRUD functionality was used during the simulation stage
- modeling module was initiated

Grade 20/20

Database Developer | 30 points

- DB and schema was successfully implemented
- Connection between SQL and Python is available
- Data is loaded
- no modifications in SQL functionality

Grade: 25/30

API Developer | 30 Points

- run.py is missing
- Requests are available, but not tested out of the package

Grade: 25/30

M2 Grade: 100/120

Milestone 3 | Tasks

Remaining tasks from M2

- Fix the file structure
- provide tests from the API
- move data one level higher

DataCamp

Complete the third chapter.

Product and Project Manager | 30 points

1. Design the final endpoints. What kind of outputs is your package going to provide?

2. Communicate the outputs with the team in order to help them create/modify the final classes/methods, etc.

3. Couldn't find `mkdocs` package in the requirments.txt

4. Create sample documentation using mkdocs. Once you have the final version, you'll update it. For now, you need to push to GitHub:

    - select a template
    - index.md page1 and page2 with dummy content (though you are free to provide actual documentation as well)

Data Scientist and Data Analyst | 30 points

- Create a predictive model based on the Product Manager's requirements
- Insert the outcome into the respective SQL table. (communicate with the Product Manager and DB developer in case you need extra table and/or functionality)
- Data analyst must try to interpret the model or create custom visualizations.

Database Developer | 30 points

- Based on the new/updated requirements, provide functionality in order to interact with the DB

    - API developer might need custom functionality for the final endpoints
    - Data Scientist/Analyst developer

- no modifications in sql functionality

API Developer | 30 Points

- Fix related files
- create the endpoints based on the requirements of Product Manager

Milestone 3 | Feedback

Ramaining tasks from M2

All done!

Datacamp

Done by Everyone

Product and Project Manager

- Final endpoints are provided
- Sample documentation is provided

Grade: 30/30

Data Scientist

- The predictive model is created
- The values are inserted, however the outcome wasn't directly inserted into DB.
- Data Analytics is done

Grade: 25/30

Database Developer

All done!

30/30

API Developer

All done!

30/30

Grade: 115/120

Milestone 4 | Tasks

1. **Documentation 30 points**
    - Create comprehensive documentation using **MkDocs**.
    - Each module (e.g., API, database, logger, model) should have its own dedicated page within the documentation.
    - The first page should provide a high-level overview detailing the **Problem**, **Solution**, and **Expected Outcomes.**
    - Host the completed documentation on **GitHub Pages.**
2. **README.MD 25 points**
    - The README file is also going to be the first page description in pypi.org. So make sure to make it as informative as possible.
    - mkdocs weblink
    - steps using the package
    - API GET Requests (the links which are showing up in the swagger under the each endpoint)
    - put it in `setup.py` (in order to make it available on pypi)
3. **Requirements and Environments 15 points**
    - Develop at least two requirements.txt files to manage dependencies more effectively.
        - `package_requirement.txt`
        - `docs_requirements.txt`
    - Create two separate virtual environments
    - for the main package (excluding **ipykernel** or **notebook** and other not directly related packages)
    - building the documentation
4. **Repository Management 15 points**
    - Clean up the repository to ensure it contains no extraneous files.
    - Host the main package on PyPI.
5. **Demonstration Notebook: 15 points**
    - Provide an `example.ipynb` file **outside** of the main package.
    - This notebook should demonstrate at least two scenarios where the solution is applied effectively

Milestone 4 Feedback

Documentation

- The MkDocs weblink is missing.
- The docstrings are not properly written.
- Input argument types are partially provided, for instance, in the `customer_segmentation.py` file.

Grade: 10/30

README.MD

- You haven't included the `setup.py` file, hence the package description on PyPI.org is missing.

- The MkDocs weblink is incorrect as it points to localhost.

Grade: 10/25

Repository Management

- The repository contains redundant files:
    - Multiple `.db` files
    - Multiple `schema_builder.py` files
- The package is hosted and working properly.

Grade: 12/15

Requirements and Environments

- You haven't separated environments for MkDocs and the package, resulting in unnecessary packages/dependencies being included.

Grade: 10/15

Demonstration Notebook

Done.

Grade: 15/15

M4 Grade: 57/100

Demo | 20 points

You need to introduce the product with 10 minutes.

The presentation format:

- **Slide 1:** The Problem
- **Slide 2:** Solution
- **Slide 3:** The problem solving methodology
- **Slide 4-5**: Demo
    - Anything you'd like to show
    - business case scenario 1
    - business case scenario 2

Demo Grade: 20/20

Final Grade

Grade: **317/400** (-10 points for Davit)