PMLDL Project Deliverable D1.1
Apartment condition classification from images

Rafik Hachana, Fadi Younes, Kamil Sabbagh

October 2022

1 Sprint summary

In this sprint, we have mainly focused on the Data Collection step of our project. The data needed for training our models needs to be scrapped from the Otodom and Morizon websites for real-estate rental.

Therefore, we have implemented a scraping Python tool that runs in the background, and can collect data from the websites over the span of a couple of days. The data consists mainly of images, as well as some tabular data about each apartment entry. At each iteration, the scraper saves all of the collected data in a MongoDB database instance for later usage. The scraping tool also takes advantage of the Python multithreading capabilities, by extending the base Thread class and using it to run multiple scrapers at the same time.

The main challenge we faced in the development of the scraping tool was properly parsing the website content. We needed to implement 2 different child classes (one for each website) that extend our base scraper in order to properly scrape each of the websites.

We have also deployed a MongoDB server cluster on the free Atlas service that is offered by MongoDB. We will start effectively using this database instance by the start of next sprint, when we will deploy and run the scraping tool.

Besides, we also started the deployment of scraping tool by setting up a free-tier Amazon Web Services EC2 instance. The setup has not been fully successful yet so we could not effectively use it in this sprint.

Finally, we have also started researching potential models to use for our classification task. Since we are doing image classification, the best models would be a Convolutional Neural Network. The most notable models that we found and might use in the modeling step are VGG-16, Inceptionv3, ResNet50, and EfficientNet. We have also found a study that, interestingly, used LSTM networks for apartment pictures classification.

We have pushed all of our code progress (currently only the scraping tool) to our project's GitHub repository.

2 Team members contribution

- **Fadi Younes**: Implemented the Scrapper Python classes for both of the websites, and the multithreading module to optimize the scraper.

- **Kamil Sabbagh**: Implemented the integration with the MongoDB database, also deployed the MongoDB cluser on the Atlas cloud services.

- **Rafik Hachana**: Setup of an AWS cloud instance to run the scraper, CLI interface for the tool. Started the research about potential models.

3 Next goals

- Run the scraping tool on the cloud and populate the database.
- Prepare data for training (formatting, potential augmentations, ...)
- Attempt training / fine-tuning models.