Project Name: \_\_\_\_

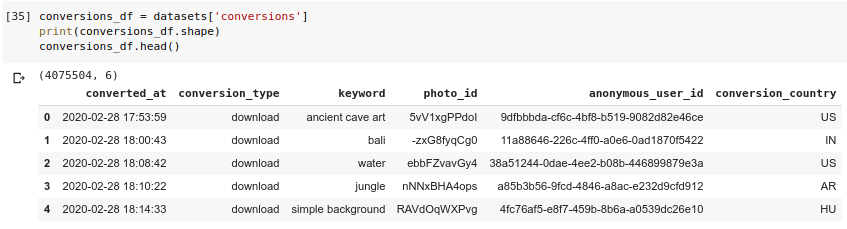
Members: \_\_\_\_\_\_\_\_\_

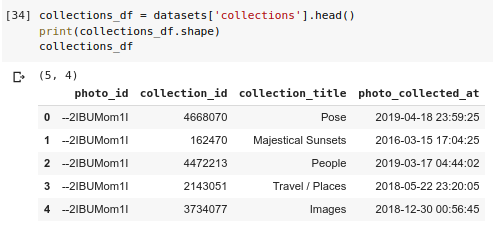
Our project goal is creating a model that classify images and label them by common tag.

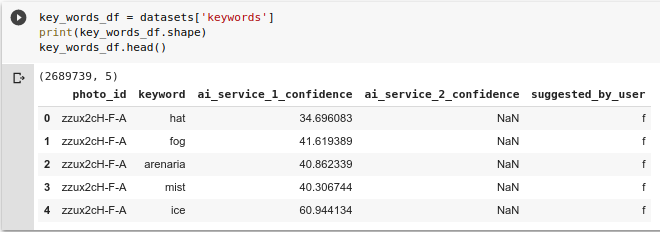
In the first stage, the model relies on images from this website: <https://unsplash.com/>

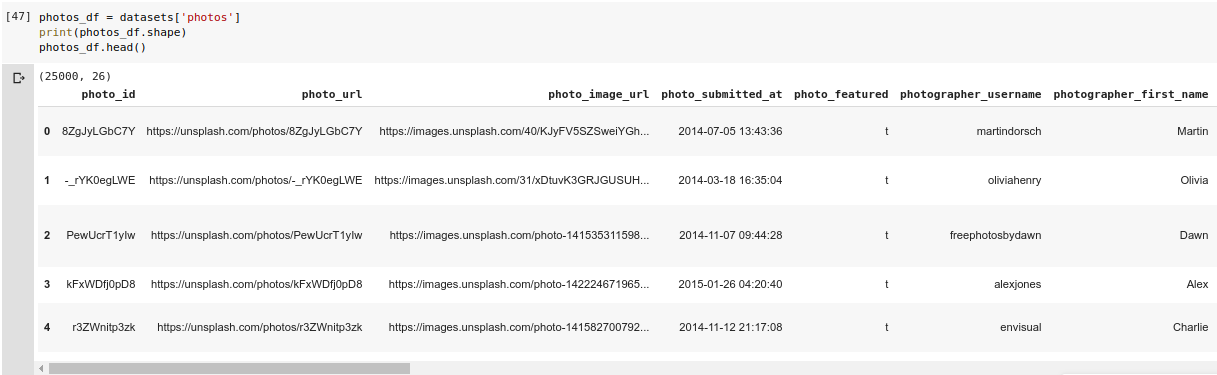
**EDA:**

The dataset contains 4 tables with details about the images, from which 2 of them are important to our needs:

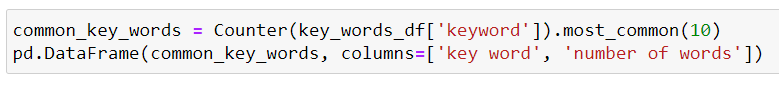


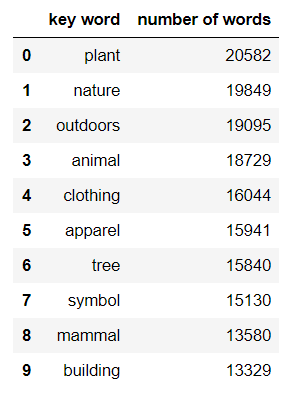






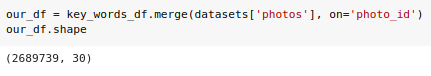
To choose the tags for the model we found the most common tags in the dataset:

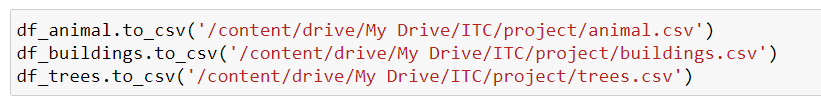


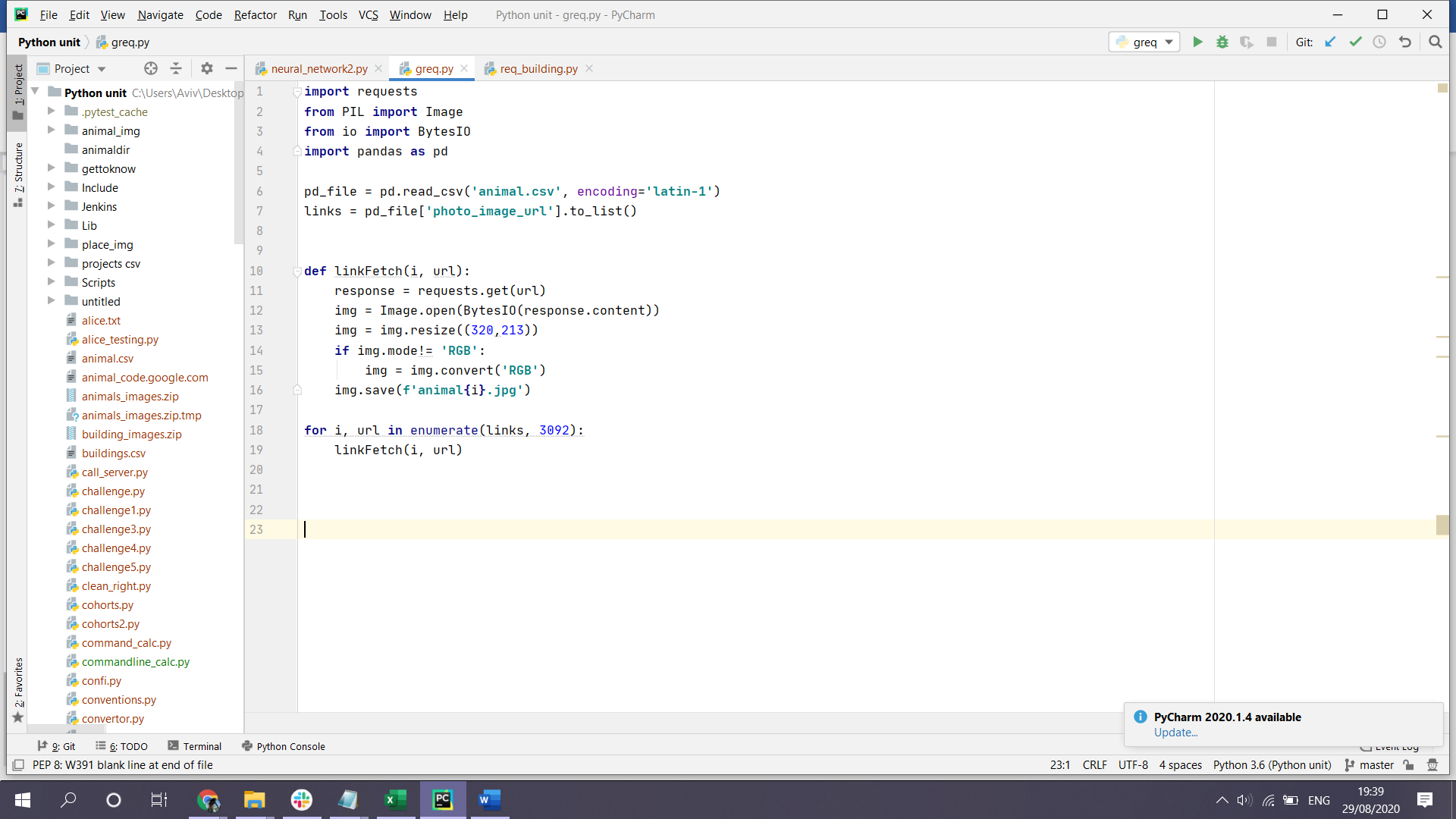


We decided to focus on three common tags that are easy to distinguish between them: trees, buildings and animal.

To get their images we created 3 files with the tag name and the photo\_image\_url columns using photo\_df and key\_words\_df tables:





Afterwards, we download the images using requests:

Since we needed to download images (by using BytesIO module on the response content) we could not use grequests. We, therefore, made three copies of the script which run simultaneously and downloaded images from each class.

The download procedure included reading the csv file using pandas and extracting the links of the images. We built a function which sends the request to the server and then converts the content into an image using the PIL library. We resized all images to reduce the load on our model.

In addition, from time to time there was an image crashing the programme as it was not in RGB mode. We converted all non-compliant images to RGB and saved them as a jpg file, using the class and image number as a dynamic file name generator.

Next steps

we started with what we perceived as the most 3 distinguished tags - animal, person, tree. Our first next step is to add more tags from the most common tags of our dataset. we need to consider that some of the tags will be more related to each other so prediction results can be less accurate. in that stage one optional solution will be to predict multiple labels, we will consider building new loss function or just taking the results of all labels up to the Nth place (when N will be about 2-5).

we are going to try to build auto tagger for Instagram posts. first, we need to find attractive tags for Instagram posts that are widely used and scrap posts that include those tags in order to build a database. in order to choose the right tags, we need also to consider their similarity to our already predicted tags. until now our tags are categories that mostly correspond to objects or objects categories, but with posts tags like ‘YOLO’, it is not the case, as these posts can be related to a lot of very different objects. so we forecast that this will be a major issue to solve.

next, we will try to replace our last layer of the network with some of the most common or most-liked Instagram tags that we found and train the network in order to predict those tags. in this way, if we will get good results we will be able to provide the user of the social network a list of suggested popular tags to her posts that will increase her posts exposure to other users and save the effort of phrasing the right tag for an image (so much effort).

a further suggested improvement is to build LSTM model that will predict tags based on text from posts. this can use as a validation to the auto tagger in cases that we can get posts composed from image and text, or be part of a model combining the CV with the NLP to predict better tags for posts.

CP - 3

Creating csv file:

