General Instructions (updated: 28.01.2023)

Data

> UPDATE (28.01.2023.): New and combined data sets are uploaded.

- We decided to combine the data sets for each taxonomic level, so that every team works with the same initial data set.
- You should pick one taxonomic level and only work with that `.csv` file. For example, if you chose to work with *Species*, you only need *train_combined_Species.csv* file.

Data Set Preparation

- You should create a Python script called *scriptname.py* that contains the procedure you used to preprocess the data. This procedure should be the one you used to create the final DataFrame you used to train your model. This procedure may include any kind of preprocessing of the data (merging two or more columns, deletingg some columns, PCA on some columns, *etc.*).
- The name of the script should be `scriptname.py` where *scriptname* is consisted of the taxonomic level you used and the first names of your team. For example, if John, Anne, and Marie are in the same team and they used the *Family* data set, your *scriptname* should be `Family_JohnAnneMarie.py`. The taxonomic level starts with the capital letter!
- We will use this script to preprocess the test data, so that your model just works when we load it. **The script should not contain functions - we must be able to run it from the terminal**. We will place the test data in the same directory as the script and run it with `python scriptname.py`.

The script must contain the following:

```
import os
import pandas as pd

path_input_data = os.path.join('.', 'test_combined_Order.csv')  # If we
trained our model using the Order taxonomic level. If not, it can be for
example test_combined_Species.csv

df_data = pd.read_csv(path_input_data, delimiter=',')
df_data.drop(df_data.columns[0], axis=1, inplace=True)  # We drop the index
column

############################
# You do the preprocessing here
############################

path_output_data = path_input_data
df_data.to_csv(path_output_data)
```

Important !

Please test your script by placing the training data in the same folder as the script, and changing the `path_input_data` parameter with `path_input_data = os.path.join('.', 'train_combined_Order.csv')`. It should work without any problems when you run `python scriptname.py` (*e.g.*,, `python Family_JohnAnneMarie.py`).

ML Model

1. Exclude the first column for training. You can use all other columns. The first column is an index column with an empty name.
2. You can use any model from the *scikit-learn* or the *tensorflow* library.
3. When models are trained and ready to be evaluated by us, follow this procedure:
    1. If your model comes from the *scikit-learn* library use `joblib.dump(model_variable, 'model_name.joblib')` where *model_name* is consisted of the taxonomic level you used and the first names of your team. For example, if John, Anne, and Marie are in the same team and they used the *Family* data set, your export command should look like `joblib.dump(model_variable, 'Familiy_JohnAnneMarie'.joblib')`. You can find more information about the *joblib* package here: https://scikit-learn.org/stable/model_persistence.html.
    2. If your model comes from the *tensorflow* library use `tensorflow.keras.Model.save('model_name.h5')` where *model_name* is consisted of the taxonomic level you used and the first names of your team. For example, if John, Anne, and Marie are in the same team and they used the *Family* data set, your export command should look like `tensorflow.keras.Model.save('Family_JohnAnneMarie.h5')`. You can find more information about the *tensorflow.keras.Model.save* method here: https://www.tensorflow.org/tutorials/keras/save_and_load#save_the_entire_model.

How to send the results

1. Your e-mail should contain the subject in the following format: *Hackathon Results - team_name*. For example: *Hackathon Results - JohnAnneMarie* for the team that consists of John, Anne, and Marie.
2. You should create a zip file named **JohnAnneMarie.zip** which contains the saved model (*.joblib*, or *.h5*), and the data set preparation Python script (*.py*).
3. You should attach the zipped file **JohnAnneMarie.zip** and send it to `aleksandar.anzel@uni-marburg.de`.

General Info

The deadline for submission is **12.02.2023.** at **23:59**.