

# Homework #4. Improve and Deploy Model & Report

## Tasks

1. Train models.
  - a. Choose and train 3 different [ML models](#).  
[https://scikit-learn.org/stable/supervised\\_learning.html](https://scikit-learn.org/stable/supervised_learning.html)  
It should not be Linear Regression, Logistic Regression or LinearSVC.
  - b. Write the motivation why did you chose these models.
  - c. Save all 3 models as pickle files.  
<https://docs.python.org/3/library/pickle.html>
  - d. Calculate confusion matrices for the 3 final tuned models. Explain what do metrics mean.
  - e. What are the top 20 features for each of your top-3 models? Draw them and their weights.
  - f. Choose the best model. Motivate your decision on how you define the “top” model?
2. Tune hyperparameters for the top model of your choice:  
[https://scikit-learn.org/stable/modules/grid\\_search.html](https://scikit-learn.org/stable/modules/grid_search.html)
3. Upload the top model in the Google Drive folder -  
[https://drive.google.com/drive/folders/1FMosTkxFCqO7rM-admBv5YQwUSK0s-zl?usp=drive\\_link](https://drive.google.com/drive/folders/1FMosTkxFCqO7rM-admBv5YQwUSK0s-zl?usp=drive_link).  
**Naming:** <command\_id>\_\_<model\_name>\_\_<version>.pkl  
**Example:** 1\_\_random\_forest\_\_v1.pkl  
Models with the wrong naming will be fined (-1pt).
4. Write a list of TODO to improve your top model: What can you improve? What new features can you generate? What improvement can be made to the final model?
5. Implement the capability(script) to make predictions for each region for the next 12 hours and save them. This script must be applied every hour (use cron for this purpose).  
<https://serverspace.io/support/help/automate-tasks-with-cron-ubuntu-20-04/>  
<https://www.digitalocean.com/community/tutorials/how-to-use-cron-to-automate-tasks-ubuntu-1804>
6. Deploy your system as SaaS. The idea is to have an interface (API) to interact with your prediction system.
7. Implement endpoint to update forecast (executes the script from task #1 and saves results in the file (or database)).
8. Implement an endpoint to give users an alarm forecast for the next 12 hour for the selected region (should be part of the request body) or for all regions (if region is not selected or is equal “all”).  
The resulting data structure should be:

```
{
```

```

"last_model_train_time": "2023-02-01T13:15:30Z", #time when model was retrained last time
"last_prediciotn_time": "2023-04-01T13:15:30Z", #time when the prediction was updated
<your meta information>
"regions_forecast": {
  "Donetsk": {
    "12:00":true,
    ...
    "00:00":false
  },
  "Lviv":
  {
    "12:00":true,
    ...
    "00:00":false
  }
}
}

```

9. Record a video of how your SaaS works:
  - a. With your server dashboard (ec2 by default)
  - b. Postman requests to your SaaS:
    - i. Make prediction
    - ii. Show new prediction file
    - iii. Show how the endpoint to give prediction to the user works (reads from the file and returns the result).

Video must be uploaded on youtube (can be available only via link).

10. Write a short report with:
  - a. Title page - H/W name, team, teacher, date.
  - b. Links: git, youtube.
  - c. Deployment process.
  - d. How did you implement the model's regular prediction capability.

11. A presentation about your project:
  - a. First slide - Project name, team, teacher, date.
  - b. Agenda
  - c. Introduction (why it's important)
  - d. Problem statement
  - e. System design
  - f. Details about each module
  - g. Demo
  - h. Git link.

12. Write a short description and instructions for your solution in GitHub README file.

13. \*(Optional task, +5pt) Create capability (script) to retrain the model with the historical dataset + new data.

### Expected Outcome

1. PDF report (task #6).
2. Video recording (1-2 min).
3. Postman collection.
4. PDF presentation.

### Deadline

15/04/2024, 23:59 (max: 15pt)