FoML Hackathon Submission Guidelines

1 Formatting Instructions

Kindly read the following instructions carefully. Your submission will not be evaluated if incorrectly formatted eg. with uppercases, spelling mistakes. Please submit a single zip file `rollno_foml24_hackathon.zip`. Do not submit tar files. The zip file must contain a folder `rollno_foml24_hackathon` which contains two files – a **python** file `rollno_foml24_hackathon.py` and a requirements file `requirements.txt`.

You can convert your notebook into a py file using `jupyter nbconvert --to python cs21resch01004_foml24_hackathon.ipynb`. **Ensure that the data analysis plots are commented out. Your python file should not output any plots. See more in Section 3**.

Your submission zip should have the following structure.

```
cs21resch01004_foml24_hackathon.zip
|--cs21resch01004_foml24_hackathon
    |--cs21resch01004_foml24_hackathon.py
    |--requirements.txt
```

- Ensure your rollno is lowercase in every file. Follow all formatting instructions.
- You can assume `numpy, scipy, pandas, scikit-learn, catboost, xgboost, lightgbm` are present with us. If you want a particular version of the libraries, or any additional libraries, add them to `requirements.txt` file, for example:

```
tensorflow==2.18.0
pytorch==2.1.0
```

- Do not have any additional .py helper files. Your submission must contain exactly one python file which will be executed.

2 Code Instructions

Your python code must run on a new `private_test.csv` file and must write the predictions, **in the same format as a Kaggle submission**, to an output csv file, whose path will be given as a command line argument.

- We will copy over the train file so that the `train.csv` file will be present in the same folder as your python file. You should read your training data simply as `train_df = pd.read_csv('./train.csv')`. **Do not hard code absolute paths of your personal machine or your submission will not run. Do not use a different name for the training file other than train.csv.**

- Your code will take a private test file path as input from the command line via the argument `--test-file`. Your predictions for the private test file must be written to an output file whose path will be provided via a command line argument `--predictions-file`.

Below is an **example** of how your python file should look. It is important to take the command line arguments exactly as described, but the rest of the code structure is up to you.

```
import argparse

def my_train_fn(*args, **kwargs):
    train_df = pd.read_csv("./train.csv")
    ###some code###
    return trained_model

def make_predictions(model, test_fname, predictions_fname):
    #### this is an example ###
    test = pd.read_csv(test_fname)
    test_X = test[features].to_numpy()
    preds = model.predict(test_X)
    test_uid = test[["UID"]].copy()
    test_uid["Target"] = preds.reshape(-1)
    test_uid.to_csv(predictions_fname, index=False)

if __name__=="__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("--test-file", type=str, help='file path of test.csv')
    parser.add_argument("--predictions-file", type=str, help='save path of
        predictions')
    args = parser.parse_args()

    #perform training
    model = my_train_fn()
    #evaluation
    make_predictions(model, args.test_file, args.predictions_file)
```

Again, the structure above is an example of how it can be done. Ensure is that your code reads the train data from `train.csv` present in the folder iteself, and runs on a private test file provided by us.

3 Testing You Code

Test your submission before uploading. The output.csv file is the only thing that will be considered.
Copy over existing `test.csv` into your home directory `~`. Unzip your submission folder to a new path, say `~/Downloads/cs21resch01004_foml24_hackathon`, and `cd` into it. Now copy the `train.csv` file into this folder. While still inside the folder, execute `python cs21resch01004_foml24_hackathon.py --test-file ~/test.csv --predictions-file ~/output.csv`. Check for a `output.csv` in your home directory.

- Test your submission on a Ubuntu machine. Use a friend's machine if you only have Windows. Do not test on Colab.
- Do not plot anything, or have any popups. Your code must execute without needing any prompting and exit without errors. The only output we will consider is the output.csv file.
- Ensure that your code runs on the CPU only (no GPU).