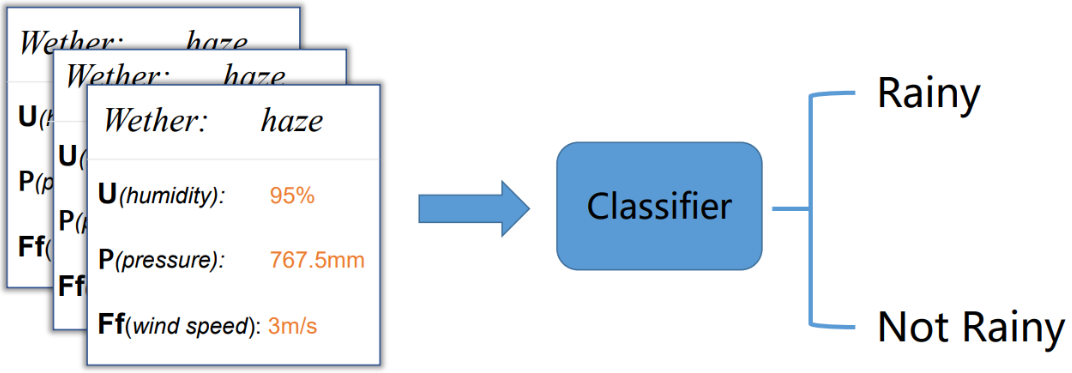
Description

- Weather prediction plays a crucial role in various aspects of daily life and planning. In this machine learning project, you are tasked with the challenge of implementing a **classification** algorithm that categorizes the day as either "Rainy" or "Not Rainy".



Dataset

- You can download the training data from here. The dataset contains 40774 weather records, each record has 25 attributes. The meaning of each attribute is shown in the following table.

Attribute	Meaning
Time Stamp	The time stamp at this record (we relabel the years from 0001-0018)
T	Atmospheric temperature at 2 meters above the ground
Po	Atmospheric pressure at meteorological station level
P	Atmospheric pressure at mean sea level
Pa	Atmospheric pressure change over the past 3 hours
U	Relative humidity at 2 meters above the ground
DD	Wind direction at 10 to 12 meters above ground in the last 10 minutes
Ff	Average wind speed at 10 to 12 meters above ground in the last 10 minutes
ff10	Maximum gusts at 10 to 12 meters above ground in the last 10 minutes
ff3	Maximum gusts at 10 to 12 meters above ground between two observations
N	Total cloud amount
WW	Current weather condition reported by the weather station
W	Past weather between observations
Tn	Lowest temperature in the past 12 hours
Tx	Highest temperature in the past 12 hours
Cl	Stratocumulus, stratus, and nimbostratus clouds

Attribute	Meaning
Nh	Amount of cloud layer C1 observed
H	Height of the base of the lowest cloud layer
Cm	Altostratus, altocumulus, and nimbostratus clouds
Ch	Cirrus, cirrocumulus, and cirrostratus clouds
VV	Horizontal visibility
Td	Dew point temperature
tR	Time to reach a specified amount of rainfall
RRR	Amount of rainfall

- **Notice**:
  1. For some training samples, some features and even labels are missing. Before building a model, you may want to clean the data or think about how to use the data whose features or labels are missing.
  2. Each record in the dataset represents weather conditions at a specific time during a day. Your objective is to predict if it rained at any time during the entire day. A day should be classified as 'rainy' if there is rainfall in any of the time segments. For example, if one segment shows no rain but others do, the day counts as having rain.
- Your model will be tested on the testing dataset. In the testing dataset, labels “Rainy”, and “Not Rainy” are set to 0 and 1, respectively. The attributes of testing sample are the same as training sample.

Requirements

- Do **NOT** use any autograd tool or any optimization tool from machine learning packages. You are supposed to implement your algorithm from scratch. For example, if you want to use a neural network, you are expected to implement both forward and backward passes. You can use the packages in the Whitelist. TAs will update the Whitelist if your requirements are reasonable.
- You can work as a team with no more than **three** members in total. Please list the percentage of each member’s contribution in your report, e.g., {San Zhang: 30%, Si Li: 35%, Wu Wang: 35%}.
- We define a default base class called `PB21000000` in `PB21000000.py`. You are supposed to implement your algorithm in `[your student ID]` directory. We provide an example code here. For detailed requirements, please refer to the comments in our code.
- You are supposed to send a package named `[your student ID].zip` , which contains the `[your student ID]` directory organized as follows to ml2023fall_ustc@163.com.

```
[your student ID]
├── main.py
├── [your student ID]-report.pdf
└── ... (your code and model)
```

For a teamwork, please use the team leader's student ID in the package name and submit the package by your team leader.

- Remember to save the trained model. You are supposed to send your trained model to the aforementioned e-mail address.
- Please submit a detailed report. The report should include all the details of your projects, e.g., the implementations, the experimental settings and the analysis of your results.

For TA's test

- The run command that we use to run your submitted python script is : **python main.py --dataset=/home/hyliu/ML_Project/testing_dataset.xls**. The required output is the **f1_score** of your model.
- In the testing phase, your sunmitted python scripy is regarded as a black-box process that should satisify the above run command and output requirments. Notably, you should load your model path in the submitted python file in advance with name **'/home/hyliu/ML_Project/your_model_path'**.

Grading

- The full points = min(Base score (up to 20pts) + Bonus (up to 5pts), 20pts).
- The base score is determined by the Macro F1-score, precision and recall evaluated by TAs' code.
- The bonus involves three aspects as follows.
  - Your insights on the data and task.
  - The novelty of your approach, which should be highlighted in your report.
  - The readability of your code and report. Please make them easy to follow.

System Requirements

- We will evaluate your model on a GeForce RTX 3090Ti (about 24G memory) under Ubuntu 18.04 system. Please limit the size of your model to avoid OOM.

Due Day

- Team leaders should inform the TAs about your team members before **23:59 PM, November 28, 2023**.
- Please submit your report, code and trained model before **23:59 PM, January 19, 2024**.
- No late submissions will be accepted.