

Data science HW3

Department of Computer Science
National Tsing Hua University (NTHU)
Hsinchu, Taiwan

Date: April 13th, 2021
TA Email: tfg10232338@gmail.com

HW3

- Description
- How to submit and choose predictions
- Baseline method
- Hints

Kaggle

- HW3 will be held on Kaggle
 - **Please register a Kaggle account first**
- A platform of
 - Machine learning competition
 - Sharing dataset
- <https://zh.wikipedia.org/wiki/Kaggle>

The Kaggle logo, featuring the word "kaggle" in a lowercase, blue, sans-serif font.

HW3



- HW3 Kaggle link
 - <https://www.kaggle.com/t/8257d4321c074ef1b587535c2c775883>
- Deadline: **2021/05/11 23:59** (4 weeks)
- We will **use the result on Kaggle to score** this homework
 - *No need to hand in any files on iLMS*
 - **Remember to fill your Kaggle name in the google form**
https://docs.google.com/spreadsheets/d/1cg-BPX6gKrcEXLbWzhnxKsYkZNGXXfPN6n7JZHlvN_U/edit?usp=sharing

Problem description

- **Supervised binary classification problem**
- Given a data set
 - Training set with label
 - Testing set without
- You need to predict the labels of testing data

Dataset description

- The dataset is **transformed** from real weather observations dataset
- 16 numeric features, 5 nominal features, 1 label
 - *Numeric feature are nonlinear transformed*
 - *About 20% data become missing value*
- Our dataset label is '**Weather**'

Output format

- For each testing instance, there is a unique id
- Output your prediction to csv file with the following format and submit to kaggle

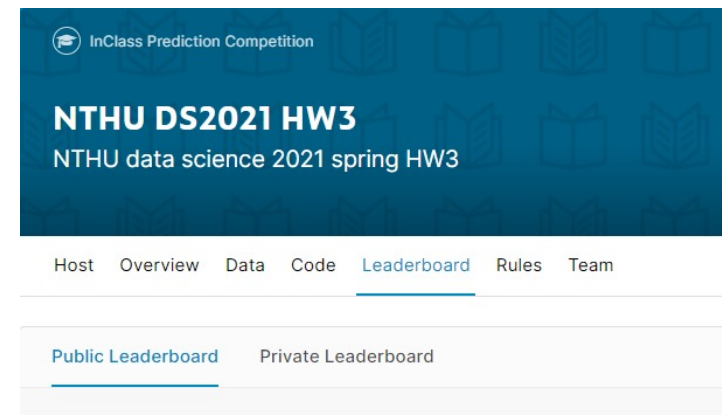
Remember to output the first line

- Id, Weather
- Id1, Weather 1
- Id2, Weather 2
- ...

1	Id	Weather
2	0	0
3	1	0
4	2	0
5	3	0
6	4	1
7	5	0
8	6	0

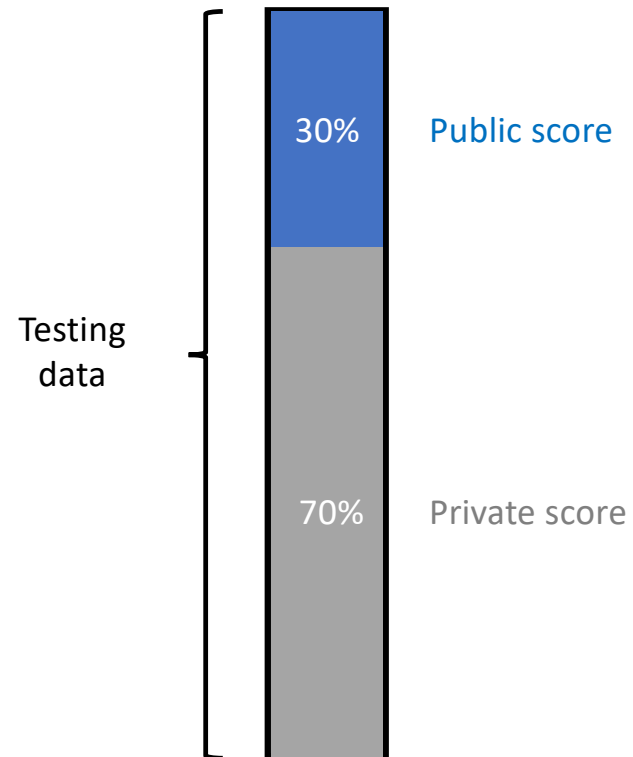
Evaluation

- We use F1-score
 - $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$
- There are two leaderboards on Kaggle
 - **Public**
 - Can be seen during competition
 - **Private**
 - Can be seen after competition



Public and Private leaderboard

- **Public** (Can be seen during competition)
 - 30% testing data
 - For reference
- **Private** (Can be seen after competition)
 - the other 70%
 - **Use this result for final scoring**

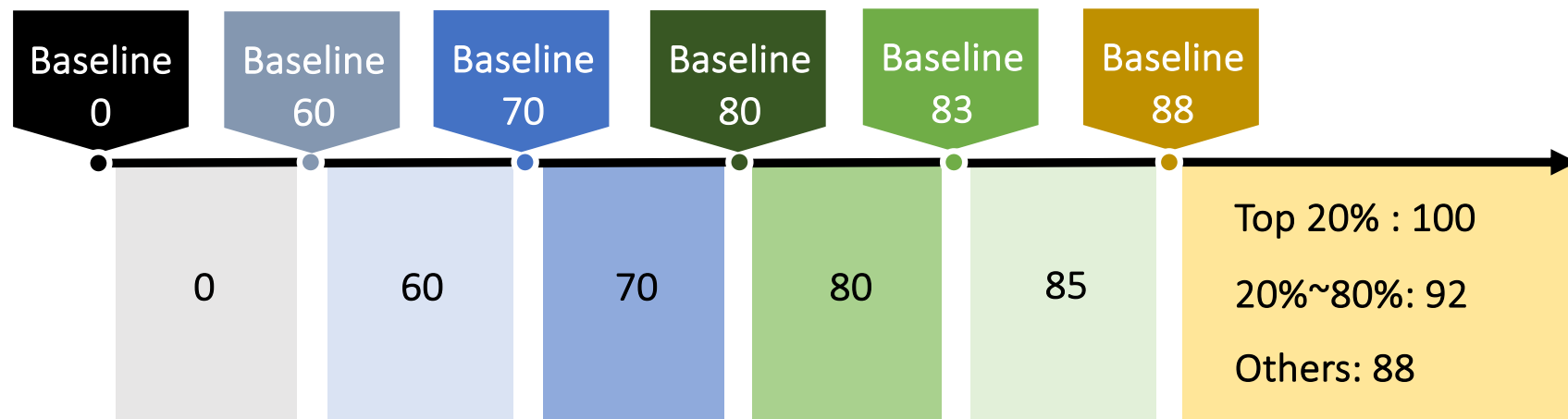


Scoring

- Use *private leaderboard result* for final scoring
- Baseline scores
 - We will score according to given 7 baseline scores

	Public	Private
Baseline 90	0.43261	0.42595
Baseline 85	0.41335	0.42502
Baseline 80	0.38176	0.38108
Baseline 70	0.33909	0.35150
Baseline 60	0.28220	0.28336
Baseline 0	0.26584	0.25060

Scoring



- You will get **0**, if your private score is between *baseline 0* and *baseline 60*
- You will get **60**, if your private score is between *baseline 60* and *baseline 70*
- You will get **70**, if your private score is between *baseline 70* and *baseline 80*
- And so on

Scoring

- Baseline scores
 - There are benchmarks on the leaderboard for reference

#	Team Name	Notebook	Team Members	Score ?	Entries	Last
📍	baseline 88			0.43261		
📍	baseline 83			0.41335		
📍	baseline 80			0.38176		
📍	baseline 70			0.33909		
📍	baseline 60			0.28383		
📍	baseline 0			0.26584		
📍	sampleSubmission			0.15112		

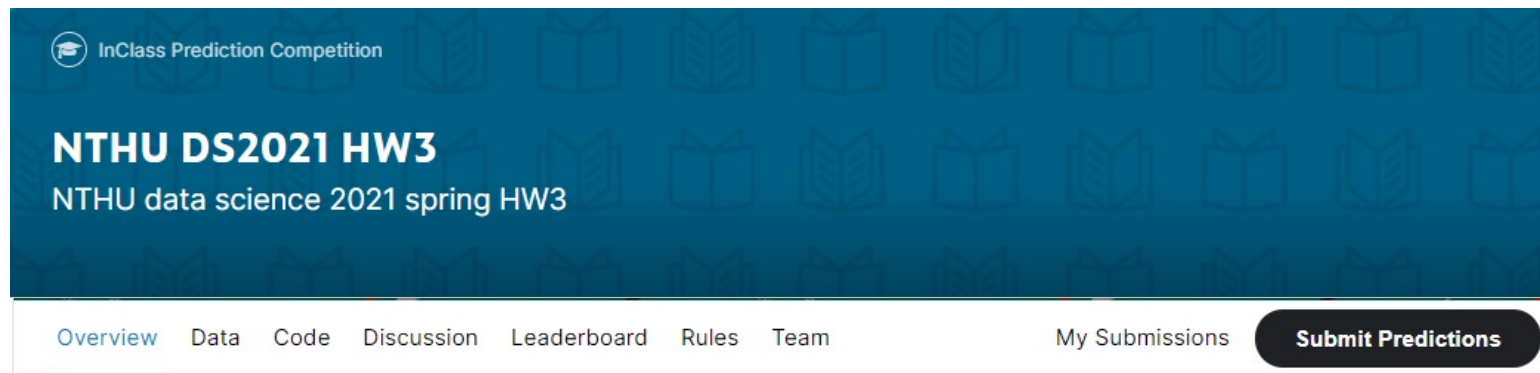
Other rules

- You can submit 15 times per day
- You can choose 4 predictions for final scoring
 - Kaggle will use the best one to be your final result

How to submit and choose predictions


How to submit

- Click '***Submit Predictions***' button on the navigation bar



How to submit

Step 1
Upload submission file

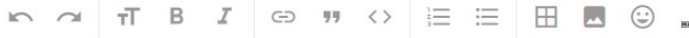


Upload your answer csv file here

File Format
Your submission should be in CSV format. You can upload this in a zip/gz/rar/7z archive, if you prefer.

Number of Predictions
We expect the solution file to have 214200 prediction rows. This file should have a header row. Please see sample submission file on the [data page](#).


Step 2
Describe submission



Briefly describe your submission

You can write some description about the answer csv file

Make Submission




Click to submit

Choose predictions for final scoring

- You can see all your submissions in ***'My Submissions'***

Overview	Data	Code	Discussion	Leaderboard	Rules	Team	My Submissions	Late Submission	
							myAns.csv	0.26102	<input type="checkbox"/>
							a year ago by		
							test		



- **Remember to choose 4 predictions before the deadline**

Baseline method

Baseline method

- We provide a simple baseline method code for your reference
 - **Baseline 0**
- The steps in baseline are as below
 - Read training/testing data
 - Drop columns which are not numeric features
 - Fill missing value
 - Train a *decision tree* classifier
 - Output prediction

Baseline 0 method

- Read training/testing data

```
[1] ▶ ▶≡ MI
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

[2] ▶ ▶≡ MI
# 為了處理方便，把 'train.csv' 和 'test.csv' 合併起來，'test.csv' 的 Weather 欄位用 0 補起來。
df = pd.read_csv('train.csv')
df_test = pd.read_csv('test.csv')
df_test['Weather'] = np.zeros((len(df_test),))

# 以 train_end_idx 作為 'train.csv' 和 'test.csv' 分界列，
train_end_idx = len(df)
df = pd.concat([df, df_test], sort=False)
```

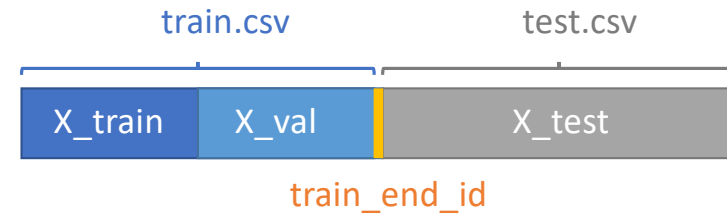
Baseline 0 method

- Drop columns which are not numeric features
- Fill missing value

```
▶ ▶≡ M↓  
# 將非數值欄位拿掉  
df = df.drop(columns = [col for col in df.columns if df[col].dtype == np.object])  
  
# 將 missing value 補 0  
df = df.fillna(0)
```

Baseline 0 method

- Split dataset



```
[4] ▶ ▶ Ml  
from sklearn.model_selection import train_test_split  
  
X_train, X_val, y_train, y_val = train_test_split(  
    df.drop(columns = ['Weather']).values[:train_end_idx, :],  
    df['Weather'].values[:train_end_idx], test_size=0.5)  
  
X_test = df.drop(columns = ['Weather']).values[train_end_idx:, :]
```

Baseline 0 method

- Train a decision tree classifier and output prediction

```
[5] ▶ ▶ ML
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, f1_score

#train tree model
model = DecisionTreeClassifier()
model.fit(X_train,y_train)

#predict
y_pred_decision = model.predict(X_val)
print('Accuracy: %f' % accuracy_score(y_val, y_pred_decision))
print('f1-score: %f' % f1_score(y_val, y_pred_decision))

Accuracy: 0.837695
f1-score: 0.264122

[6] ▶ ▶ ML
ans_pred = model.predict(X_test)
df_sap = pd.DataFrame(ans_pred.astype(int), columns = ['Weather'])
df_sap.to_csv('myAns.csv', index_label = 'Id')
```

Hints

Hints

- You can try to encode features in object type
 - Some features in object type may contain important information

```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
df['Loc'] = labelencoder.fit_transform(df['Loc'])
...
```

Hints

- Fillna with median in numeric features instead of 0

```
df[i] = df[i].fillna(median)
```

- Deal with data imbalance

```
from imblearn.over_sampling import SMOTE  
sm = SMOTE(random_state=42)  
X_train,y_train = m.fit_resample(X_train,y_train)
```

Complete these may achieve the same or higher effect as the baseline 60

Hints

- Try different models
 - KNN, SVM, Logistic Regression, Random Forest ...

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
```

Finetune the model may achieve higher effect than the baseline 70 and 80

Hints

- More techniques for better performance
 - Feature selection
 - Normalization
 - Dimension reduction (PCA, TSNE)
 - Try other different models
 - ...
- We use private leaderboard as the final score
 - Use public score to choose your model is dangerous
 - It's better to perform validation

Packages you may use

- Scikit-learn
 - <https://scikit-learn.org/stable/index.html>
- Pandas
 - <https://pandas.pydata.org/pandas-docs/stable/>
- Imbalance learn (for over sampling and down sampling)
 - <https://imbalanced-learn.readthedocs.io/en/stable/>