Intern Quiz - 潛力熱門文章預測

繳交期限**: 2020/04/16 23:59:59**

繳交平台**: Greenhouse**

1.題目

在 Dcard 上，有一個很重要的版面叫熱門文章，每天會有許多的使用者會看這個列表來得知 Dcard 站上最火熱的討論話題是什麼。但身為做資料的人，我們也很想知道哪些文章是有潛力上熱門的，如果我們在推薦的時候考量這個因素進來也許能夠更快的讓使用者知道這是一篇好的文章。所以在這個作業裡面，我們希望能夠根據一些資料，來預測某一篇文章是不是有機會上到熱門，希望大家會享受這個挑戰 XD。如果有任何關於題目的疑問的話，請回信給我們，我們會視情況回答你的問題～謝謝！

熱門文章定義

為了簡化問題複雜度，我們目前訂為在文章發出的 36 小時內愛心數 >= 1000 就是熱門文章。實際測試的時候我們會去計算 36 小時內的某篇文章的愛心數是否超過 1000 來當做答案或是預測的基準。

---

2. 你需要做到的事

- Report (不限格式)
  - 怎麼使用你們的 code
  - 方法以及為什麼要這樣做
  - Evaluate 在我們提供的 testing data 的結果
  - 實驗觀察
- Code quality
  - 基本上官方無法產生最終結果的話視同未完成
  - 程式碼可讀性
  - code structure
  - 不接受 jupyter notebook 當作提交的程式，這個作業的目的之一其實是讓大家體驗實際專案在運行的時候會遇到的狀況。
- Evaluation metrics
  - 在做 offline evaluation 的時候只會使用每篇文章前 10 小時的資料當作是預測資料
  - 以 f1-score 為主

---

3. 回傳檔案要求

程式語言

由於我們日常開發是 python，所以希望這個專案的程式也是使用 python 做開發

格式

我們對這個 Task 的程式會有一些要求，所有的程式的 package structure 請自行規劃。但希望繳交上來的程式可以符合下面的規格以利測試：

- training
  - 最外層用 train.py 包著
    - 實際會執行 `python train.py {database_host} {model_filepath}`
    - example: `python train.py localhost:8080 ./model.h5`
- predict
  - predict.py
    - 實際會執行 `python predict.py {database_host} {model_filepath} {output_filepath}`
      - 請告訴我們你的 `model_filepath` 放在哪裡
    - example: `python predict.py localhost:8080 ./model.h5 ./sample_output.csv`
  - 你的程式最終在預測的時候要能夠做到下列兩件事情
    - 從資料庫讀資料，資料的格式跟下一個 section 說的一樣，最後做 judgement 的時候我們會用自己做的另外的資料做測試。
      - 實際上我們會取另一個資料庫的 `xxx_test` tables 當測試集。所以 `predict.py` 裡面請吃這些 table 當成是你的程式的 input。
    - 輸出成 CSV 格式，裡面有兩個 column 如下，需要輸出 header (請參照我們附上的 sample_output.csv)
      - post_key: string type
      - is_trending: bool type

最後必須回傳的資料

在完成作業後，你回傳的作業內容必須至少包含下列的 1 - 4 這四個檔案，未包含視同未完成。

1. `Report.pdf`
2. `train.py`
3. `predict.py`
4. `requirement.txt` 或 `pipfile`
5. (Optional) 如果 predict 需要 model file 請務必附上 (我們不會幫你 train)，並在 `Report.pdf` 裡說明如何執行

---

4. 資料庫

我們會提供資料庫供大家使用，連線資訊如下:

- host: X
- port: X
- user: X
- password: X
- database: X

如果不太熟悉怎麼用 python 連到資料庫的，可以參照 appendix

---

5. Data

我們會用 PostgresSQL 提供資料給大家，裡面將包含

1. training / testing data
    a. 提供到文章發佈後 10 小時的資料
    b. 文章三十六小時後的愛心數在 posts 裡面的 like_count_36_hour
2. 在做完作業後，你可以 evaluate 在我們提供的 testing data 上面，並將結果記錄在 `Report.pdf`

Posts

文章的資訊

Table name

- `posts_train`
- `posts_test`

Schema

- post_key
- created_at_hour
- like_count_36_hour

Example

表示文章在 `2019-06-01` 早上五點到六點之間發文，36 小時內累積愛心數量 115 個

post_key	created_at_hour	like_count_36_hour
ob4ef005-9956-4d5e-bbe9-f413b0a8ef87	2019-06-01 05:00:00	115

Post Share

文章每小時內的分享次數

Table name

- `post_shared_train`
- `post_shared_test`

Schema

- post_key
- created_at_hour
- count

Example

表示 post_key `ob4ef005-9956-4d5e-bbe9-f413b0a8ef87` 在 `2019-06-01` 早上五點到六點間被分享 1 次

post_key	created_at_hour	count
ob4ef005-9956-4d5e-bbe9-f413b0a8ef87	2019-06-01 05:00:00	1

Post Comment

文章每小時內的留言次數

Table name

- `post_comment_created_train`
- `post_comment_created_test`

Schema

- post_key
- created_at_hour
- count

Example

表示 post_key `ob4ef005-9956-4d5e-bbe9-f413b0a8ef87` 在 `2019-06-01` 早上五點到六點間有 3 個留言

post_key	created_at_hour	count
ob4ef005-9956-4d5e-bbe9-f413b0a8ef87	2019-06-01 05:00:00	3

Post Like

文章每小時內被愛心的次數

Table name

- `post_liked_train`
- `post_liked_test`

Schema

- post_key
- created_at_hour
- count

Example

表示 post_key `ob4ef005-9956-4d5e-bbe9-f413b0a8ef87` 在 `2019-06-01` 早上五點到六點間有 `5` 個愛心

post_key	created_at_hour	count
ob4ef005-9956-4d5e-bbe9-f413b0a8ef87	2019-06-01 05:00:00	5

Post Collected

文章每小時內被收藏的次數

Table name

- `post_collected_train`
- `post_collected_test`

Schema

- post_key
- created_at_hour
- count

Example

表示 post_key `ob4ef005-9956-4d5e-bbe9-f413b0a8ef87` 在 `2019-06-01` 早上五點到六點間被收藏 12 次

post_key	created_at_hour	count
ob4ef005-9956-4d5e-bbe9-f413b0a8ef87	2019-06-01 05:00:00	12

---

6. Appendix

要怎麼連上 **database**

如果不太熟悉 database 的人，可以考慮整合使用下面的 postgres_connector function 來連到 database 取資料

```
import pandas as pd
import sqlalchemy

# Connector function
def postgres_connector(host, port, database, user, password=None):
    user_info = user if password is None else user + ':' + password
    # example:    postgresql://federer:grandestslam@localhost:5432/tennis
    url = 'postgres://%s@%s:%d/%s' % (user_info, host, port, database)
    return sqlalchemy.create_engine(url, client_encoding='utf-8')

# Get connect engine
engine = postgres_connector(
    "X",
    X,
    "X",
    "X",
    "X"
)

# Query example
query = """
SELECT *
FROM xxx
WHERE
  ooo < xxx.abc
"""
pd.read_sql(query, engine)
```