Cloud Computing and Cyber Security 2020 FALL

Term Project Report

Project Title

學生使用線上平台學習情況之分析 - 以均一教育平台為例

Project GitHub URL

https://github.com/MorrisWCC/CloudFinal

Team members

- R07922158 王俊中
- R09922114 蘇泰宇
- R09922152 沈培文

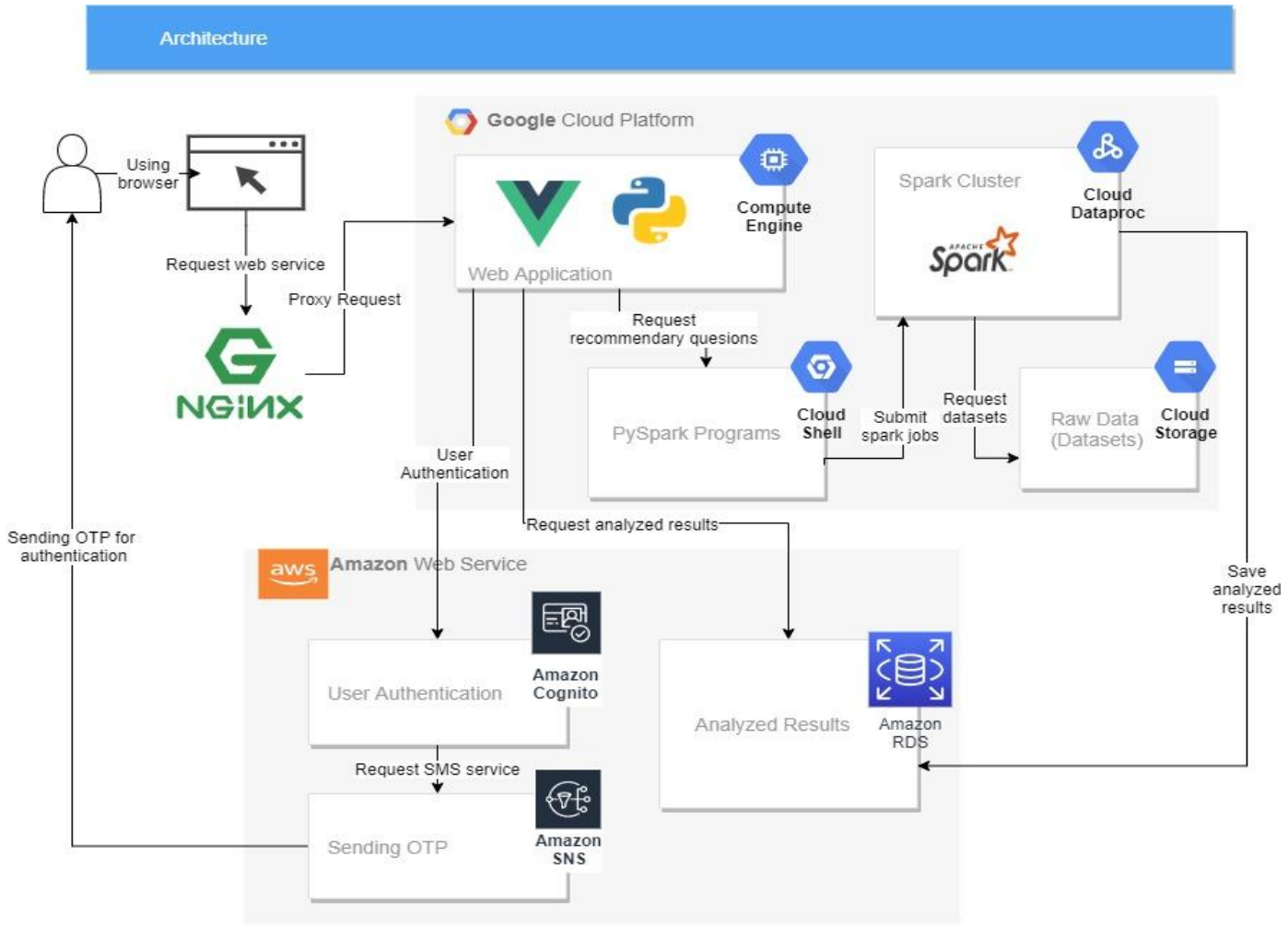
Abstract

線上學習平台在當今社會已經成為極為重要的學習資源之一，我們希望能夠透過分析這些學生在線上學習平台的 log 檔案為學生創造更好的個性化學習體驗。

本專案專注於整體資料的分析，例如: 對於特定年級、特定學科、特定主題的統計分析，希望透過這些分析數據找出有利於學生學習的資訊。此資料集的大小約 2.8 GB，雖然還未達到真正的大數據，但已經是在學期間較少能夠接觸到的數量級，希望能夠透過此專案的實作更加了解這些資料處理的平台。

我們運用了公有雲平台(GCP、AWS)架設我們的服務，從 Web服務、計算平台架設、至使用者身分認證皆是透過公有雲的服務協助完成，此外我們的服務亦考量到安全性之問題，在整個網站皆使用 Https加密協定，同時在登入系統的部分，我們透過 2FA 的機制確保使用者的帳號安全，並且讓使用者能夠使用 OAuth 的方式登入我們的服務。

Architecture



Approach

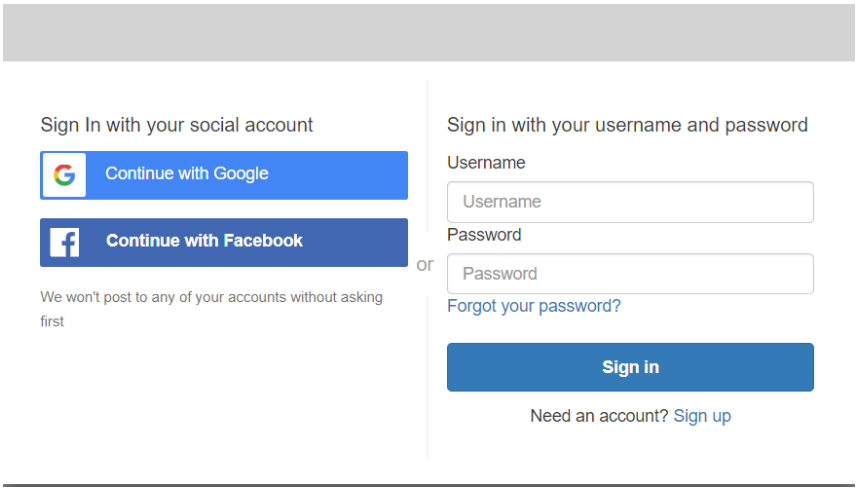
- Dataset
  - Junyi Academy Online Learning Activity Dataset from Kaggle
    - User information、Problem information、Practicing logs
  - Dataset csv files size are more than 2.8 GB
  - More than 16 million records in the log file
    - Comparing with homework 3 dataset, which contains only 1500 records.

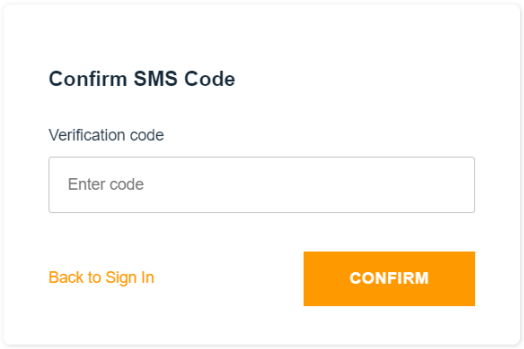
- Tech Stacks
    - Cloud Services
        - Google Cloud Platform: Dataproc、Storage、Compute Engine
        - Amazon Web Service: RDS、Cognito、SNS、Amplify
    - Back-end
        - Python Flask、Nginx、PySpark、SparkML
    - Front-end
        - Vue.js with Vuetify
    - Computing Platform
        - Spark cluster
    - Database
        - MySQL
- Proposed Analysis
    - Total Accuracy of all problems
    - Accuracy of the each problem
    - Accuracy of the different difficulties problems
    - Active students
    - Activities status of different level school students
        - Weekday
        - Weekend
    - Practicing problems of the each student
    - Student distribution
    - Students learning status by grade and month
- Proposed Recommender System
    - According to a student information and a problem information to predict the possible accuracy, and we recommend the student the suitable problems by possible accuracy.
- Comparison with Python Pandas
    - Compared with execution time for the previous analysis.

Results

- HTTPS

- Login System with OAuth and 2FA



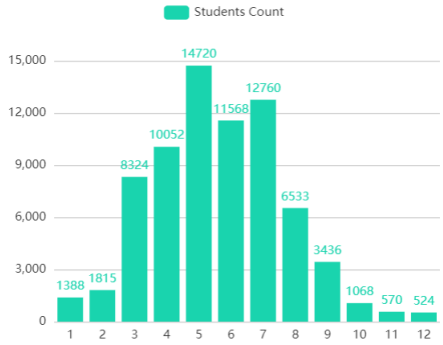


- Dataset Analysis Results

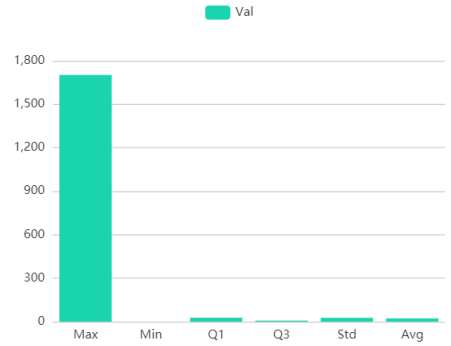
Analytics Results

Total accuracy of execises: 0.70373

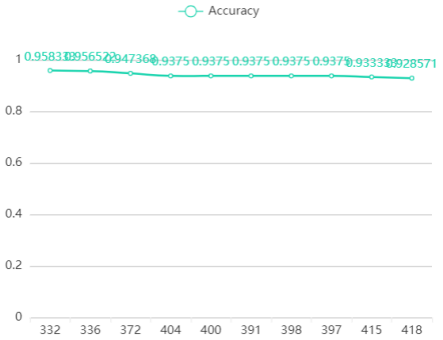
Students Distribution



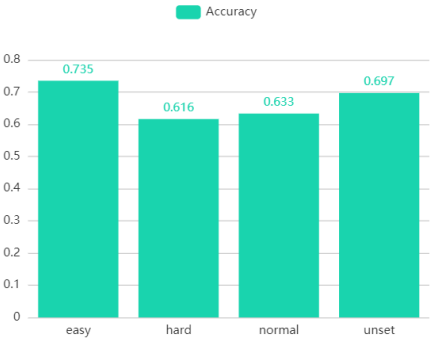
Solved Problem by Students



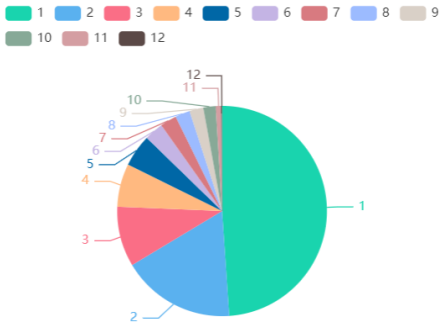
Accuracy of Each Problem



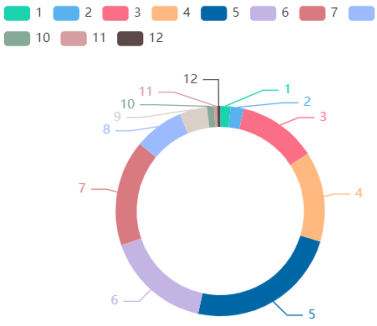
Accuracy of Different Difficulty

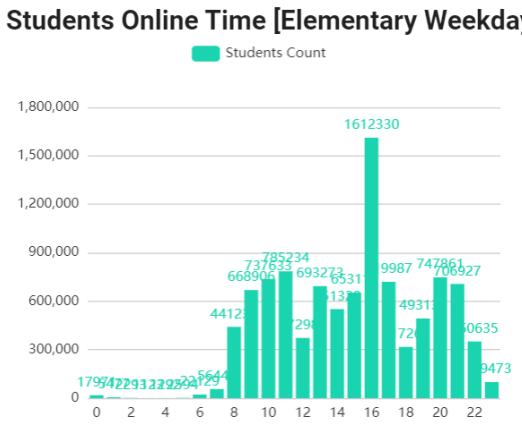


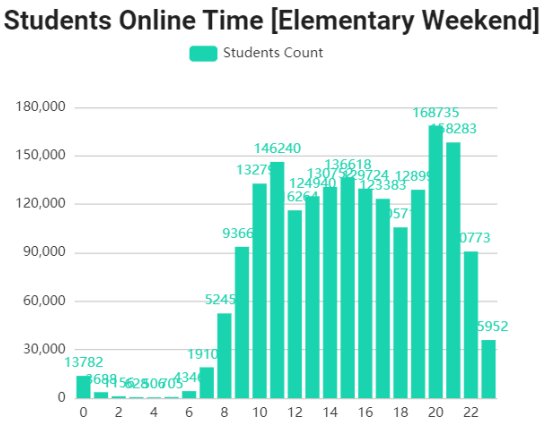
Active Students



Students Learning Status by Grade (2019-01)

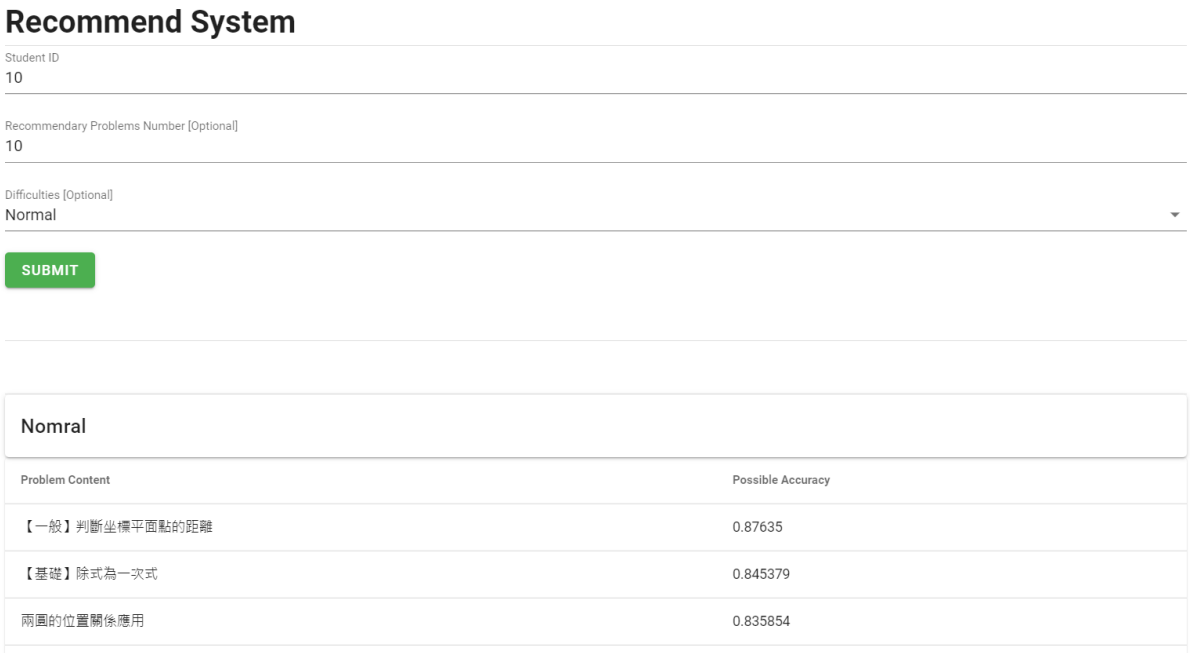






For more information about Analysis, please refer to the demo.

- Recommendation System



- Comparison with Python Pandas

We found that Spark speeds up the analysis at least 100 times than Python Pandas except loading raw data.

Discussion

以下討論以及心得融入我們各團隊成員之感想，由於是團隊專題，就不再細分各團隊成員。

首先在討論題目過程時，由於我們都缺乏處理大資料的經驗，甚至都認為會不會1600萬這個數量級的資料在受限的 Spark Cluster 會花費許多執行時間，然而在此時，實驗結果證明了這個數量級的資料對 Spark 這種平台來說微不足道。而我們也利用現在大家廣泛用來進行資料處理的 Python library, Pandas，試著去執行一樣的分析，最後發現兩者在一樣的處理方式下，Spark 在執行時間至少都快了兩個數量級，完全顯示了 Spark 這個平台在資料處理有多具有威力。

我們的專題主要可分成四大部分，以下將分點列向的去討論

1. 基礎平台架設 (Spark Cluster 、Virtual Machine、使用者認證管理.....)

   TL;DR: 公有雲協助了我們快速地架設所需要的 infrastructures，同時也提供了好用的服務減輕開發者的負擔。

   我們在這次的專案中深刻的體會到公有雲所提供的服務對開發者來說是多大的福音，也許對我們來說，架設 Virtual Machine 並不難，透過 Docker 來佈署服務也不是難事，但像是架設一個 Spark Cluster 或是 對 Database 的權限管理確是要花上非常多的時間，且隨著計算平台的更新會遇到的 Bugs 也不會少。而公有雲替我們省下了許多時間，尤其在專案中用的 Spark Cluster，利用 GCP Dataproc 只需要幾個按鈕就可以架設好，也很輕易地能利用 API 或者是 Cloud shell 進行 sumbit job的行為。

   除此之外，我們這次更使用到了 AWS Cognito 的服務，透過這個服務協助我們管理使用者的帳號，更是減輕了我們保管使用者帳號密碼以及對不同使用者登入管理的負擔。同時 AWS 也提供了 Amplify 的套件，能夠在目前主流的前端框架上，快速的將 Cognito 的服務融入在既有的專案之中。

2. 網頁服務 (Front-end、Back-end 、Database、安全性相關設定)

   TL;DR: 嘗試了不熟悉的前端框架，也透過 OpenSSL 實現 HTTPS，同時透過 2FA 的方式去保護使用者的帳號安全。體驗 OAuth 架設的過程，並對OAuth 有更深的認識。

   在 Front-end 我們選擇比較不熟悉但是在目前為主流的 Vue.js，雖然導致開發時間增加，但能夠學習現代前端網頁的開發技術實在十分值得。而我們選擇將服務架設在 EC2上也省去了處理服務如何被其他人存取的問題，有良好的 security group 能夠讓我們去設定防火牆、同時也不用利用像是 ngrok之類的工具將我們的服務開放至網路上。然而在安全上的考量，不管是利用 OpenSSL 去進行簽章使得我們的網頁服務能夠以 HTTPS 的方式連線，利用 2FA 這個現在很常見的方式保護使用者帳號安全也是我們在之前未嚐實現過的功能。而我們也提供 OAuth 的方式登入，體驗了與各種主流平台設定 Redirect URL 以及設定認證模式的過程。

3. Spark 計算 (with PySpark)

   TL;DR: 遇到大數量級的資料時，調用 API 的選擇至關重要，會影響到處理時間非常多。

   在這部份我們遇到最困難的就是在撰寫 PySpark 的程式時會發現有些差不多難度的分析，在執行時間上居然差異非常多。最後才發現有些 API 在執行時間上會造成很大的影響。因此在處理資料的演算法方面需要去思考怎樣才能夠達到最佳的效益。

4. 推薦系統的架設 (with Spark ML)

   Spark ML 提供了良好的 API 能夠讓我們使用，並設計 Pipeline去進行訓練模型。我們透過根據學生的資料、題目的資料去判斷該學生對於某題目的答對率為何，再透過答對率去推薦該學生適合的題目。

Conclusion

整體來說，在這個專案中我們嘗試將許多常見的功能加入我們的專案中，並在建構這些功能時學習到了許多技能以及對不同的功能、架構有更深入的了解。美中不足的是，我們所拿到的資料其實並不足夠完整，開放在 Kaggle 上的資料僅有去識別化後的資料，所以導致我們有許多想做的功能無法在實務上完成 (根據更多使用者的資料去進行推薦、根據更多題目的類型去分析使用者的學習狀況、能夠將此網頁服務連結至均一教育平台)。

而在使用公有雲的過程中，也確切地感受到了當所有東西都上雲時在團隊內分享以及共享服務有多容易，再也不需要去處理繁瑣的細節，只需要將權限設定正確即可。

同時在分工上，由於每個人處理不一樣的部分，有各自的需求，在溝通上就必須做得很完整，才能避免做一些沒必要的白工，或是需要不停地來回修改，也是在這專題中學習到很多的地方。

Reference

- Dataset from Kaggle:

  https://www.kaggle.com/junyiacademy/learning-activity-public-dataset-by-junyi-academy?fbclid=IwAR3Iv5Jmb4kFAOF6tSTpPMQ-LISxmAQRcsLez6aIYI_SAwDLpN3I46eZ26w

- PySpark documents: https://spark.apache.org/docs/latest/api/python/index.html

- Google Cloud Platform documents: https://cloud.google.com/docs

- Amazon Web Service documents: https://docs.aws.amazon.com/