資料科學 HW4 NBA2Vec

數據所(二) Q56084098 鄭宜崴

任務: 找某些數據/問題，通過修改後的 Word2vec 學習 OOO Embedding，並進行一些評估

構想

NBA2Vec，將 NBA 球員變成向量

1. 收集 NBA 球員資料
2. 將每個球員視為 word，每個球隊視為 sentence
3. 將同一個球隊中的球員，互相視為 context
4. 將不同球隊的球員視為 negative sample
5. 使用 word2vec 學習每個球員的 embedding

一、資料收集與處理

1. **資料下載:**

   從 NBAstuffer 網站下載到 4 個檔案，為2017 ~ 2020 的 NBA 球員資料，每一年份為一個 excel 檔案，每一列為一個球員在當年度季賽中的表現數據，共有 28 個 columns 如下:

   FULL NAME, TEAM, POS, AGE, GP, MPG, MIN%, USG%, TOr, FTA, FT%, 2PA, 2P%, 3PA, 3P%, eFG%, TS%, PPG, RPG, TRB%, APG, AST%, SPG, BPG, TOPG, VI, ORTG, DRTG

2. **手動資料前處理:**

   我們只需要 FULL NAME(球員全名)、TEAM(當年所屬隊伍)，其餘 columns 都先手動刪除，因此得到四個 csv 檔，即 2017、2018、2019、2020 季賽的球員數據，每一個 row 僅有 FULL NAME 與 TEAM

3. **程式資料前處理:**

   接著用 Python 進行資料前處理，依據每個隊伍名稱，將每年曾經待過的球員放如同一個列表中，視為一個 Sequence，比如 2017 年勇士隊，有以下 17 名球員:

   Andre Iguodala,Chris Boucher,Damian Jones,David West,Draymond Green,JaVale McGee,Jordan Bell,Kevin Durant,Kevon Looney,Klay

Thompson,Nick Young,Omri Casspi,Patrick McCaw,Quinn Cook,Shaun Livingston,Stephen Curry,Zaza Pachulia

因此，每一年的每一個球隊，都會形成一個 Sequence，NBA 總共 30 個球隊，四年總共會得到 120 個 Sequences，其中，Sequence 最長長度為 28，為 2018 年的灰熊隊，代表該隊伍在當年曾經有 28 個不同球員待過; 而 Sequence 最短長度為 13，為 2020 年的鵜鶘隊，代表該隊伍在當年僅有 13 個不同球員待過。

4. **Negative Sampling:**

   接著使用 tf.keras.preprocessing.sequence.skipgrams 自動產生 Positive pair 和 Negative pair

   - Positive pair: 曾經出現在同一隊的球員
   - Negative pair: 不曾出現在同一隊的球員

   其中，我將 skipgrams function 的 window_size 設定為前一步驟所找出的 Sequence 最長長度: 28，如此一來即可保證每一個隊伍在進行 sliding window 取 context 時，同一隊伍中的所有成員都會互為 Positive pair，且不同隊伍的球員之間才有機會互為 Negative pair。

   此處我設定每一組 Positive pair 會搭配 5 組 Negative pair

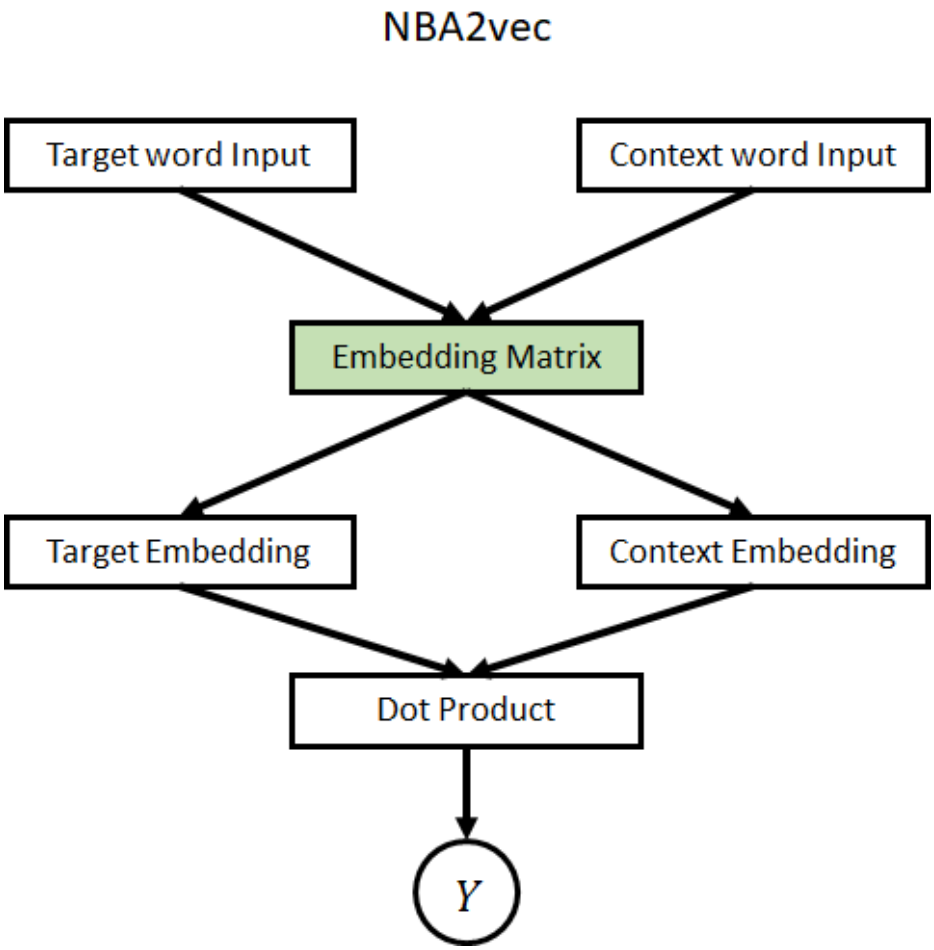
二、模型建構 NBA2vec

原始的 Word2vec 是輸入一個單字，輸出為一個多類別機率分布，代表預測出的所有詞的機率，但這種做法所需的參數量過多、計算耗時，因此後續研究提出使用 Negative Sampling 機制，將任務改為一個二分類預測，預測輸入的 word pair 是否互為上下文，若是，則 label 為 1; 若否，則 label 為 0，如此一來可大幅降低參數量、訓練所需時間。

因此，我使用 Tensorflow 建立一個 Word2vec with Negative Sampling 模型:

- 輸入層: Target word ID 和 Context word ID
- 嵌入層: Embedding Dimension 設為 32
- 輸出層: 利用 Dot Product 計算兩個詞的相似度

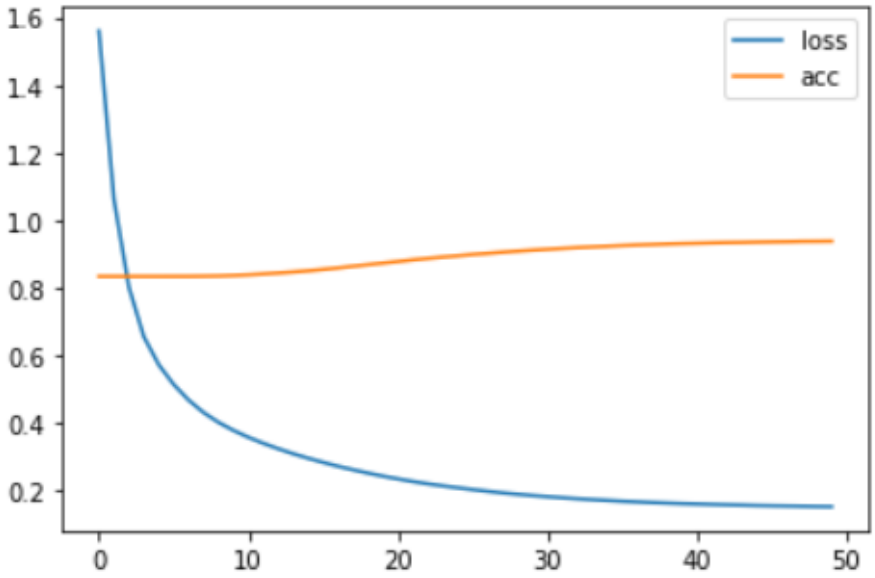
模型結構如下:




```
Model: "functional_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 1)]          0
__________________________________________________________________________________________________
input_2 (InputLayer)            [(None, 1)]          0
__________________________________________________________________________________________________
embedding (Embedding)           (None, 1, 32)        27392       input_1[0][0]
                                                                 input_2[0][0]
__________________________________________________________________________________________________
dot (Dot)                       (None, 1, 1)         0           embedding[0][0]
                                                                 embedding[1][0]
__________________________________________________________________________________________________
reshape (Reshape)               (None, 1)            0           dot[0][0]
==================================================================================================
Total params: 27,392
Trainable params: 27,392
Non-trainable params: 0
__________________________________________________________________________________________________
```

訓練歷程:

訓練 50 個 epochs，觀察 Loss 和 Accuracy 變化



三、Embedding 實驗

(1) 隨意選出幾個知名球星，計算其餘弦相似度，是否有體現出隊友相似度高、非隊友相似度低的性質

```
Cosine similarity of LeBron James to Stephen Curry is: -0.06379831582307816
Cosine similarity of LeBron James to Derrick Rose is: 0.2705168128013611
Cosine similarity of LeBron James to Anthony Davis is: 0.42189159989356995
Cosine similarity of LeBron James to Kyle Kuzma is: 0.7576419115066528
Cosine similarity of LeBron James to Kawhi Leonard is: -0.03167477995157242
Cosine similarity of LeBron James to Kevin Durant is: 0.16373300552368164
Cosine similarity of Stephen Curry to Kevin Durant is: 0.5692143440246582
Cosine similarity of Stephen Curry to Klay Thompson is: 0.8193364143371582
```

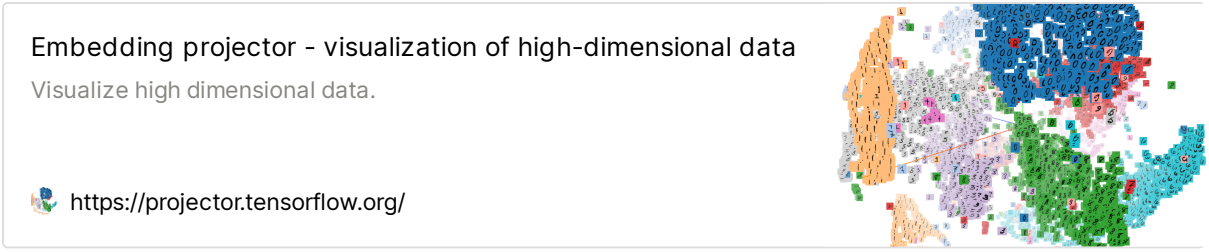
以上結果可發現:

- LeBron James 和 Stephen Curry 因為從來不同隊，因此兩人相似度非常低，約 -0.06
- 2017 年，LeBron James 還在騎士隊，和 Derrick Rose 同隊，因此相似度高了一些，約 0.27
- 2018 年，LeBron James 轉隊至湖人隊，到今年 2021 年都還在湖人隊，其中，Kyle Kuzma 從2018 ~ 2021 也都待在湖人隊，而 Anthony Davis 是 2019 年才來到湖人隊，因此 LeBron James 和 Kyle Kuzma 同隊比較久，相似度比較高，約 0.75，而和 Anthony Davis 則略低，約 0.42
- Stephen Curry 和 Klay Thompson 始終都在同一隊，因此相似度極高，約0.82

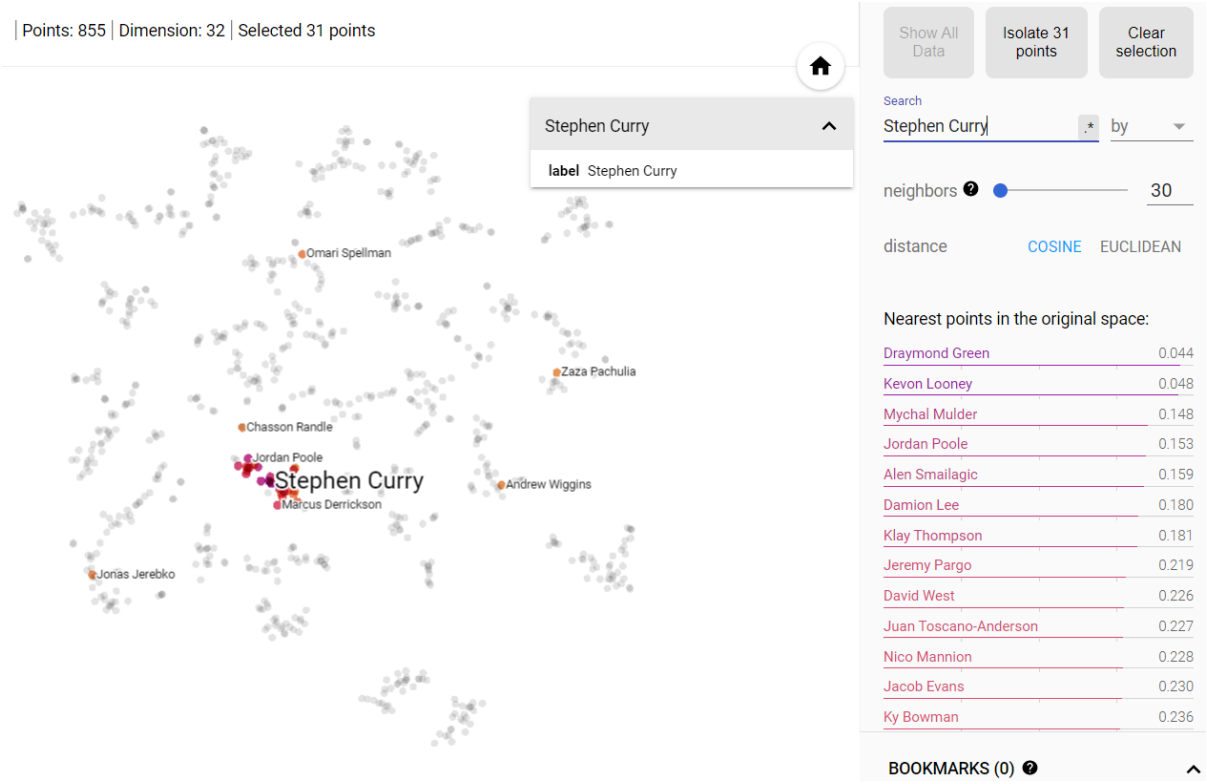
(2) 將訓練好的 Embedding，保存成為:

1. metadata.tsv : 每一列為球員姓名
2. vectors.tsv : 每一列為一個球員的 Embedding

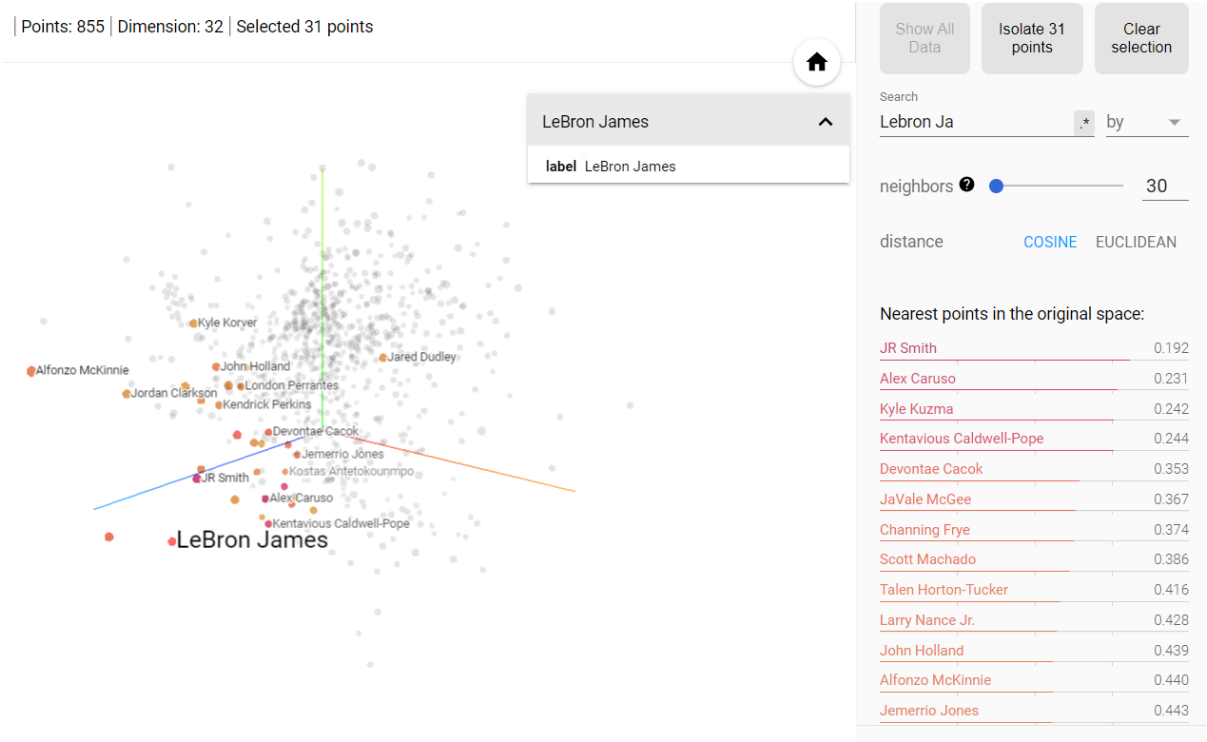
這兩個檔案可以上傳至 Google Tensorflow 團隊開發的 Embedding Projector 網站，進行線上分析:



可呈現出以下結果:

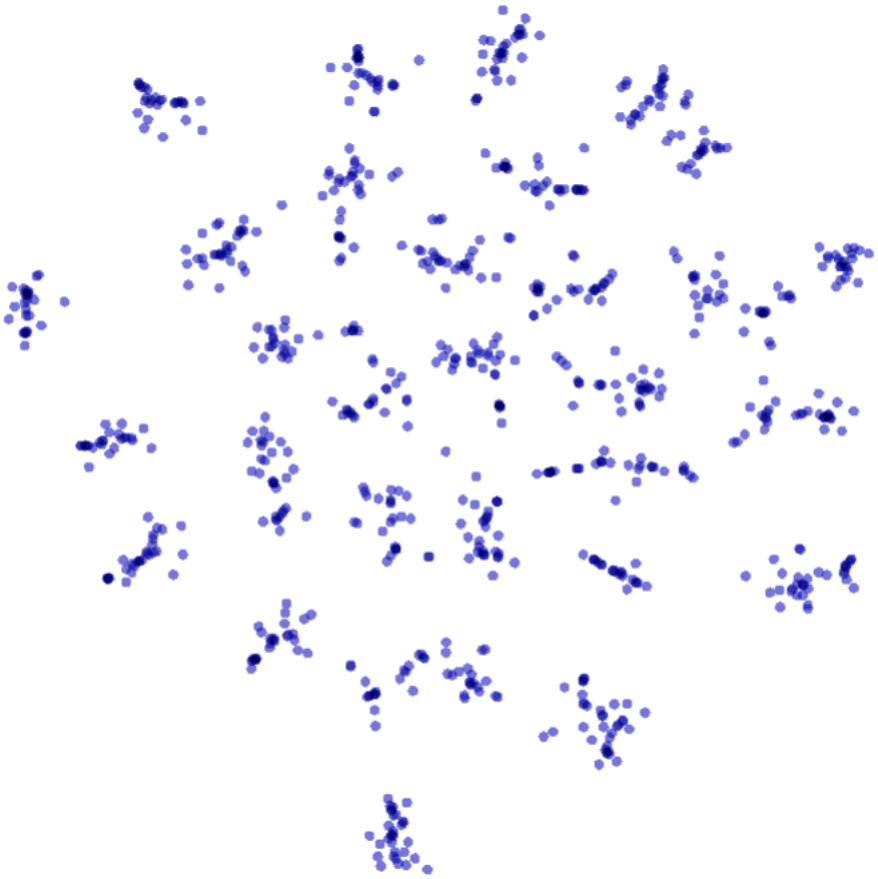


上圖可看到，與 Stephen Curry 最接近的鄰居為 Draymond Green(0.044),

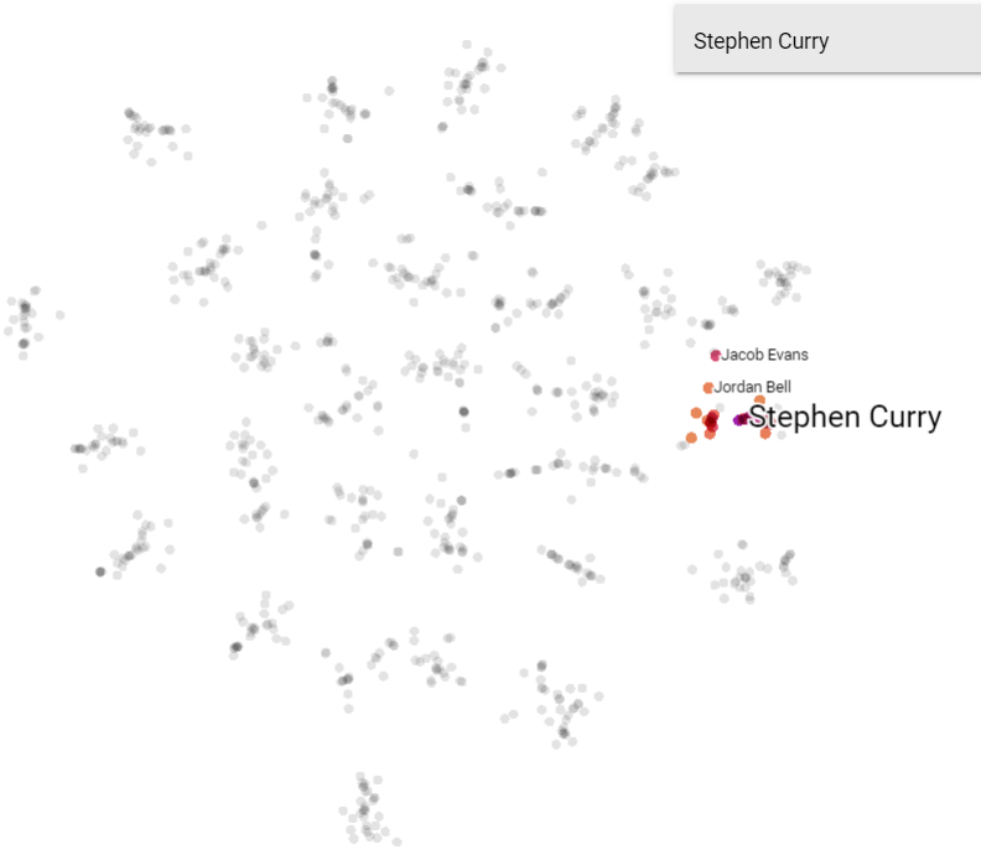


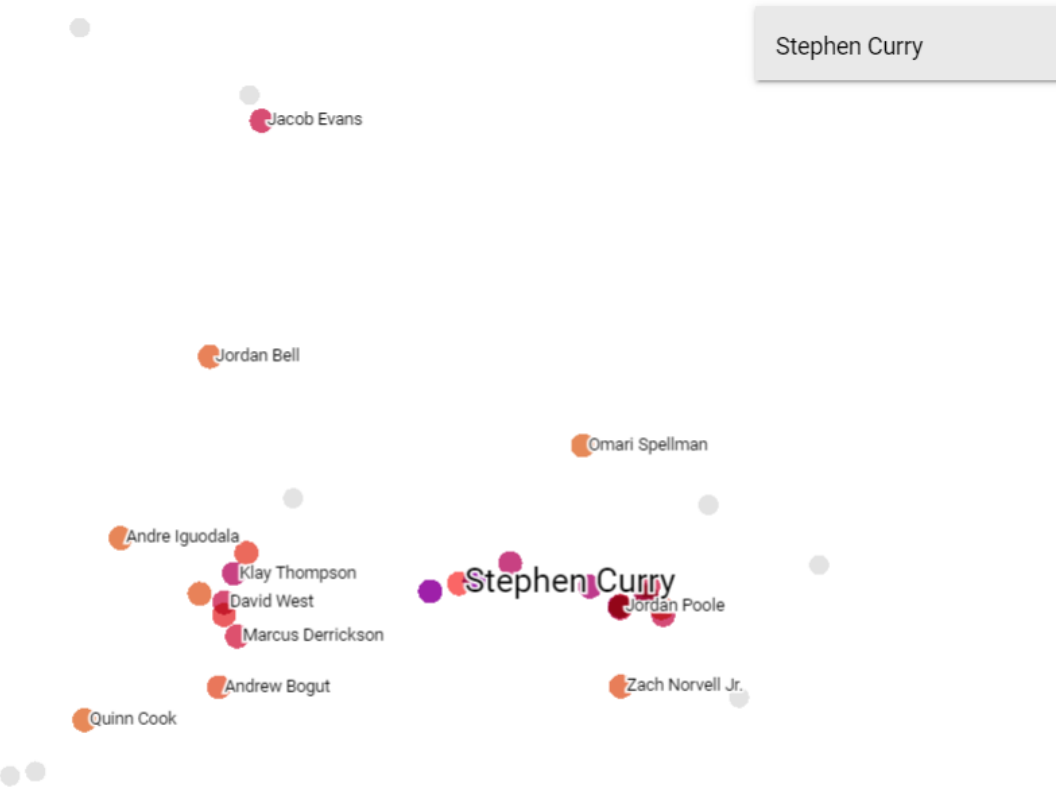
與 Lebron James 最接近的則為 JR Smith

此外，使用 T-SNE 將資料降維至 2 維，視覺化結果如下:



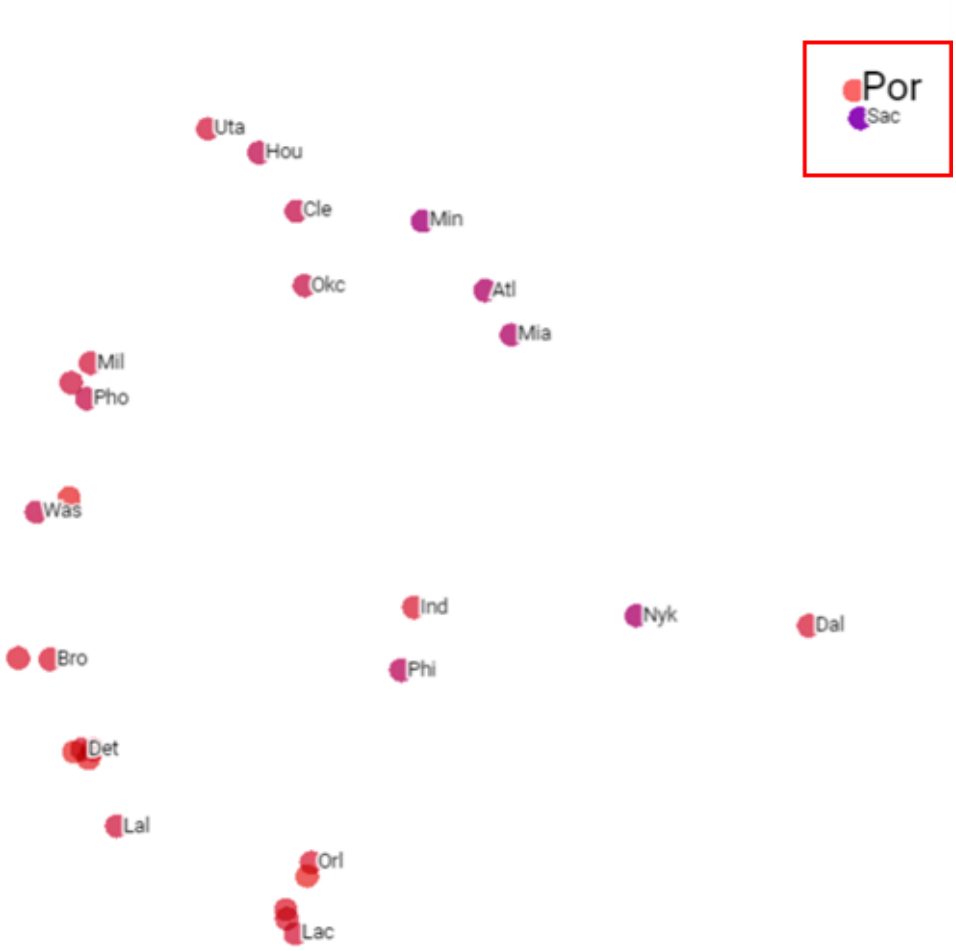
上圖可發現，分出來的這些點，正好形成了 30 個 Cluster，跟 NBA 球隊總數一樣，我們選中 Stephen Curry，再往下細看:


Stephen Curry
Jacob Evans
Jordan Bell
Stephen Curry



可發現在他周遭的，果然也都是他的隊友，因此符合預期。

接著我計算出 30 個球隊的 Embedding，觀察哪些球隊比較相近，作法是將每支球隊中所有球員的Embedding 平均，得到球隊 Embedding，同樣使用 T-SNE 後得到以下結果:



可發現右上角的 Por(拓荒者隊)和 Sac(國王隊)非常接近，正好 NBA 各球隊的地理位置，這兩支隊伍也非常靠近:

這背後的原因可能是，兩支球隊地理位置接近，因此他們之間球員流動的可能性較高，使得兩支球隊在這幾年之間擁有過的成員重複率較高。

透過以上分析，可證實所學的 Embedding 的確有捕獲到一定的球員、球隊間資訊，但因為目前每個球員之間沒有重要性之分，但事實上，某些明星球員對於球隊的代表性意義更大，未來可以考慮將不同球員的重要性也考慮進來，學習更豐富的 Embedding。

這份作業讓我學習到非常多，對於大家耳熟能詳的 Word2vec，自己親手實現一次，加深了對於此演算法的理解，並且也學習發揮創意，想出一個別人未曾做過的 Idea，從資料收集、處理，到模型實現、後續實驗都親手完成，從而獲得了大量成就感，感謝老師派了這個作業!