Entity –

FIFA 18 Ultimate Team player details – name, country, overall rating, league, position, club, nation, date of birth, etc.

Table A: 3788 tuples
Table B: 9724 tuples

Web Source 1 - https://www.futhead.com/18/players/?bin_platform=ps

- Futhead is a website which collects data regarding all the players in the FIFA 18 Ultimate Team league. FIFA 18 is a soccer video-game in which players from all around the world build teams and compete. The data collected consists of various player attributes like name, age, in-game quality statistics etc.

Web Source 2 - https://www.easports.com/fifa/ultimate-team/fut/database/results?position_secondary=LF,CF,RF,ST,LW,LM,CAM,CDM,CM,RM,RW,LWB,LB,CB,RB,RWB

- EA Sports is the official maker of the game FIFA 18. They have an official database which consists of all the in-game player attributes and statistics.

Blocker –

We used an Overlap Blocker using the attribute 'club' to initially generate a candidate set of tuples having only players which play for the same club. After debugging, to account for variation in the ways the club names are written, we used q_val = 3 and overlap_size = 2.

Then we used an Overlap Blocker on the corresponding candidate set which blocked on the player name. After debugging, we agreed that the best blocker is one with q_val = 2 and overlap_size = 2.

Number of tuple pairs = **205665**

Number of tuple pairs in sample G = **450**

Learning Method Results – Set I

The results were collected using 5-fold CV and reported values are averages.

Decision Tree –

- Precision: 0.956213
- Recall: 0.968065
- F-1: 0.961591

Random Forest –

- Precision: 0.986058
- Recall: 0.974126
- F-1: 0.980000

Support Vector Machine –

- Precision: 0.937569
- Recall: 0.987879
- F-1: 0.961644

Naïve Bayes –

- Precision: 0.992308
- Recall: 0.974126
- F-1: 0.982981

Logistic Regression –

- Precision: 0.967675
- Recall: 0.980186
- F-1: 0.973640

Linear Regression –

- Precision: 0.903675
- Recall: 0.974126
- F-1: 0.936521

Initial Best Learning Based Matcher – Naïve Bayes

Debugging –

Debugging the decision tree and the random forest models didn't lead to any insight and hence no further debugging iterations were performed.

Best Matcher After Debugging– Naïve Bayes

Learning Method Results – Set J

Decision Tree –

- Precision: 0.9677
- Recall: 0.9836
- F-1: 0.9756

Random Forest –

- Precision: 0.9375
- Recall: 0.9836
- F-1: 0.96

Support Vector Machine –

- Precision: 0.9375
- Recall: 0.9836
- F-1: 0.96

Naïve Bayes –

- Precision: 0.9836
- Recall: 0.9836
- F-1: 0.9836

Logistic Regression –

- Precision: 0.9524
- Recall: 0.9836
- F-1: 0.9677

Linear Regression –

- Precision: 0.9677
- Recall: 0.9836
- F-1: 0.9756

Final Best Learning Based Matcher – Naïve Bayes

Time Estimate –

a) To do blocking – 2 hours
b) To label the data – 1 hour (per person)
c) To find the best matcher – 3 hours

Discussion Regarding Recall –

- In order to achieve a higher recall, we could potentially use a slightly more liberal blocker so that all possible varieties of true positive tuples are retained in the candidate set.
- We could also draw and label a larger sample from the candidate set E ensuring that all possible corner cases exist in our training sample (subset of G) to train a more accurate model.

Magellan Feedback –

The Magellan tool is a well thought out and written tool with adequate documentation and code examples facilitating fast prototyping.

BUGS –

One error we faced was that on installing the package using conda, we were getting the following error –

```
import py_entitymatching as em
A = em.read_csv_metadata('/Users/yashtrivedi/cs839ps2/tutorial/futhead.csv')
A['ID'] = range(0, len(A))
em.set_key(A, 'ID')
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-1-5d46c46ec7de> in <module>()
----> 1 import py_entitymatching as em
      2 A = em.read_csv_metadata('/Users/yashtrivedi/cs839ps2/tutorial/futhead.csv')
      3 A['ID'] = range(0, len(A))
      4 em.set_key(A, 'ID')
/Users/yashtrivedi/anaconda2/lib/python2.7/site-packages/py_entitymatching/__init__.py in <module>()
     40
     41 # # blocker debugger
---> 42 from py_entitymatching.debugblocker.debugblocker import debug_blocker
     43
     44 # # blocker combiner
```

```
/Users/yashtrivedi/anaconda2/lib/python2.7/site-packages/py_entitymatching/debugblocker/debugblocker.py in <module>()
    12 import py_entitymatching.catalog.catalog_manager as cm
    13
---> 14 from py_entitymatching.debugblocker.debugblocker_cython import \
    15       debugblocker_cython, debugblocker_config_cython, debugblocker_topk_cython, debugblocker_merge_topk_cython
    16
ImportError: dlopen(/Users/yashtrivedi/anaconda2/lib/python2.7/site-packages/py_entitymatching/debugblocker/debugblocker_cython.so, 2): Symbol not found: __ZNSt11logic_errorC2EPKc
  Referenced from: /Users/yashtrivedi/anaconda2/lib/python2.7/site-packages/py_entitymatching/debugblocker/debugblocker_cython.so
  Expected in: /usr/lib/libstdc++.6.0.9.dylib
 in /Users/yashtrivedi/anaconda2/lib/python2.7/site-packages/py_entitymatching/debugblocker/debugblocker_cython.so
```

The specifications of the system we are using are –

- OS: macOS High Sierra (version 10.13.4)
- Python version: Python 2.7.14
- Conda version: conda 4.5.0

Installing the package using 'pip' and starting a Jupyter Notebook from the terminal instead of using the one in Anaconda Navigator fixed the problem.