EAS504 Project Stage 3: Matching Movie Entities from IMDB and Movie Numbers

**Shaivi Pandey** Department of Data Science University at Buffalo, New York

[shaivipa@buffalo.edu](mailto:dgriffin5@wisc.edu)

# 1 Introduction

2 In this project stage I performed entity matching between two data tables holding movie information.

3 The sources that this data was collected from is provided below, and is the same as the data sources

4 for project stage 2. Below, i provide a quicklist that can be used for quickly grading our assignment.

5 I provide some further details in later sections.

# 6 2 Quick List

7 1. **Description of entity types:** I am matching movie entities in each table to one another.

8 The entities have the attributes id, title, year, mpaa, runtime, genres, director, stars, and

9 gross.

10 2. **Description of the two tables:** The two tables I use for matching are imdb3\_neg\_nan.csv

11 and thenumbers3\_neg\_nan.csv. These tables have movie information that was originally

12 scraped from IMDB.com and thenumbers.com. These tables are also special in that they

13 have empty attributes filled in with a negative 1 if the attribute value was originally missing.

14 3. **The number of tuples per table:** Table1- imdb3\_neg\_nan.csv has 4291 tuples. Table2-

15 thenumbers3\_neg\_nan.csv has 31006 tuples.

16 4. **Description of our Blockers:** I use 3 stages of overlap blockers, and 1 final custom

17 overlap blocker that i ourselves implemented. The first blocker eliminates tuples that don’t

18 have an overlap in tokens for the movie title. This reduced out data set from 133,046,746 to

19 10,956,134 tuple pairs (about an order of magnitude). I then used another overlap blocker

20 to block the data based on year. I first had to clean our data set to ensure that all data only

21 had years and no extra characters. I also note that i tried using an attribute matching

22 blocker, but it eliminated many positive tuples. The second blocker reduced our data from

23 10,956,134 pairs to 134,228 tuple pairs (about 2 orders of magnitude). Our third blocker

24 was another overlap blocker that blocked based on mpaa rating. The total number of tuple

25 pairs was reduced from 134,228 to 27,767. Our fourth and final blocker was a blocker that

26 removed tuple pairs that ire matched based on title only because they both shared the word

27 "the". I implemented this blocker after debugging the output of the first two blockers

28 and realized that many movies have the word "the" in the name. This blocker reduced the

29 number of tuple pairs from 27,767 to 6,334, which was the final number of tuple pairs used

30 for sampling and training our learning matchers.

31 5. **Number of tuple pairs in labeled sample G:** I labeled 600 tuple pairs in G.

## 32 6. First time matchers precision and recall obtained:

33 (a) **Decision Tree:** Precision=0.992857, Recall=0.968190, F1=0.980080

34 (b) **Random Forrest:** Precision=0.992857, Recall=0.984857, F1=0.988776

35 (c) **SVM:** Precision=0.989474, Recall=0.747095, F1=0.848269

36 (d) **Linear Regression:** Precision=0.992857, Recall=0.976857, F1=0.984524

37 (e) **Logistic Regression:** Precision=0.993103, Recall=0.983333, F1=0.987796

38 (f) **Naive Bayes:** Precision=0.992857, Recall=0.976857, F1=.0.984524

39 7. **Matcher selected**: Random Forest

## 40 8. Report all debugging iterations and cross validation iterations that you performed:

41 (a) **Debugging Iteration 1:** Our first debugging iteration was focused on all of the match-

42 ers as a whole. When i initially ran our sampling and labeling stage, i sampled

43 600 instances, but only about 40 of them ire true positives. So, when i performed

44 cross validation and validation on the test set, our metrics ire still high, but our test

45 set was only 11 instances. So, i decided to implement a custom blocker removing

46 instances that overlapped with only the word "the" (which was the main set of true

47 negative instances in the sampled sets i ire getting), and performed labeling on the

48 resulting set, which gave us around 160 true instances out of the 160. After performing

49 cross validation on this data set, i achieved the performance as above, but on a larger

50 set of instances that ire more varied in their nature.

51 9. **Final best matcher that you selected:** Random Forrest, Precision=0.992857, Re-

52 call=0.984857, F1=0.988776

53 10. **Final Matcher Metrics on J:**

54 (a) **Decision Tree:** Precision=0.9565, Recall=0.9565, F1=0.9565

55 (b) **Random Forrest:** Precision=1.0, Recall=1.0, F1=1.0

56 (c) **SVM:** Precision=1.0, Recall=0.6957, F1=0.8205

57 (d) **Linear Regression:** Precision=1.0, Recall=1.0, F1=1.0

58 (e) **Logistic Regression:** Precision=1.0, Recall=1.0, F1=1.0

59 (f) **Naive Bayes:** Precision=1.0, Recall=1.0, F1=1.0

60 11. **Final best matcher Y selected, trained on I and tested on J:** Random Forrest, Preci-

61 sion=1.0, Recall=1.0, F1=1.0

62 12. **Approximate time estimates:**

63 (a) **Time to install software and learn to use it:** 3+ Hours

64 (b) **Time to do the blocking:** 2 Hours

65 (c) **Time to label the data:** 1 Hour

66 (d) **Time to find the best matcher:** 1 Hour

67 13. **BONUS:** Refer to the "Bonus" section below. I kept a running list of problems i

68 encountered and thoughts i had about Magellan in general.

# 69 3 Bonus: Notes on issues, errors, and suggestions

70 1. I used conda install -c anaconda pyqt to install the pyqt5 dependencies with anaconda 2 on

71 MacOS Sierra.

72 2. You should have some test script that checks to ensure everything is working correctly after

73 installation of all of the packages. Right now, you install everything, but you don’t know if

74 everything is working until you start trying to run code.

75 3. You might want to think about creating a "conda\_install.sh" script that will install not

76 only the py\_entitymatcher, but also the necessary dependencies, rather than having a user

77 download and install everything manually.

78 4. I would break the machine learning capability out of the py\_entitymatcher tool, and focus

79 the tool more on just the entity matching part, and feature generation. I think including

80 the machine learning, while potentially beneficial to some people who don’t know about

81 ML (But if they use the tool, they probably know ML), it puts more to maintain in the

82 py\_entitymatcher and somewhat locks down the matcher to future changes in the field of

83 ML, ML models, and other ML package capabilities. I think that the tool should maintain

84 the Unix style philosophy, which is do 1 thing, do it ill, and make it easy to compose the

85 tool with other tools (aka things like stdinput pipe handling, and output that is clean, etc...).

86 The more compact this tool, the simpler it will be to be able to use and maintain it, and

87 the more likely it will be that the industry adopts the tool. The simpler to install, use, and

88 compose with other tools it is, the more likely the chances of widespread adoption will be.

89 5. Your links on this ib page: [http://anhaidgroup.github.io/py\_entitymatching/v0.3.x/singlepage.html,](http://anhaidgroup.github.io/py_entitymatching/v0.3.x/singlepage.html)

90 for running ipy notebooks indicates that the link is a .ipynb file. But it is actually an html

91 ib page, and in that ib page there is a link to download a .json ipy notebook. So this can

92 be a bit confusing.

93 6. Your ipy notebooks are specific to the user "shaivi" that created the notebook, and are not

94 generalized. I ran into errors when running one of the notebooks bc of this.

95 7. You should make a docker file that can be run for this package.

96 8. Current anaconda compiled package for anaconda2 on MacOS Sierra "10.13.4" runs

97 into issues with error: Traceback (most recent call last): File "<stdin>", line 1, in <mod-

98 ule> File "/anaconda2/lib/python2.7/site-packages/py\_entitymatching/ init .py",

99 line 42, in <module> from py\_entitymatching.debugblocker.debugblocker

100 import debug\_blocker File "/anaconda2/lib/python2.7/site-

101 packages/py\_entitymatching/debugblocker/debugblocker.py", line 14, in

102 <module> from py\_entitymatching.debugblocker.debugblocker\_cython

103 import ImportError: dlopen(/anaconda2/lib/python2.7/site-

104 packages/py\_entitymatching/debugblocker/debugblocker\_cython.so, 2): Symbol

105 not found: ZNSt11logic\_errorC2EPKc Referenced from: /anaconda2/lib/python2.7/site-

106 packages/py\_entitymatching/debugblocker/debugblocker\_cython.so Ex-

107 pected in: /usr/lib/libstdc++.6.0.9.dylib in /anaconda2/lib/python2.7/site-

108 packages/py\_entitymatching/debugblocker/debugblocker\_cython.so

109 9. Error when installing using pip: "Cannot remove entries from nonexistent file

110 /anaconda2/lib/python2.7/site-packages/easy-install.pth" Fixed by echoing an empty

111 string to easy-install.pth file, and installing through pip without the "-U" option,

112 which caused PyPrind-2.11.1 to be upgraded to PyPrind-2.11.2, but caused the instal-

113 lation error: Successfully installed PyPrind-2.11.2 backports.functools-lru-cache-1.5

114 kiwisolver-1.0.1 matplotlib-2.2.2 py-entitymatching-0.3.0 python-dateutil-2.7.2 pytz-2018.4

115 Traceback (most recent call last): File "/anaconda2/bin/pip", line 11, in <module>

116 sys.exit(main()) File "/anaconda2/lib/python2.7/site-packages/pip/ init .py", line

117 248, in main return command.main(cmd\_args) File "/anaconda2/lib/python2.7/site-

118 packages/pip/basecommand.py", line 252, in main pip\_version\_check(session)

119 File "/anaconda2/lib/python2.7/site-packages/pip/utils/outdated.py", line

120 102, in pip\_version\_check installed\_version = get\_installed\_version("pip")

121 File "/anaconda2/lib/python2.7/site-packages/pip/utils/ init .py", line 838,

122 in get\_installed\_version working\_set = pkg\_resources.WorkingSet() File

123 "/anaconda2/lib/python2.7/site-packages/pip/\_vendor/pkg\_resources/ init .py",

124

line 644, in init self.add\_entry(entry) File "/anaconda2/lib/python2.7/site-

125

packages/pip/\_vendor/pkg\_resources/ init .py", line 700, in add\_entry

for

126 dist in find\_distributions(entry, True): File "/anaconda2/lib/python2.7/site-

127 packages/pip/\_vendor/pkg\_resources/ init .py", line 1949, in find\_eggs\_in\_zip

128 if metadata.has\_metadata(’PKG-INFO’): File "/anaconda2/lib/python2.7/site-

129 packages/pip/\_vendor/pkg\_resources/ init .py", line 1463, in has\_metadata

130 return self.egg\_info and self.\_has(self.\_fn(self.egg\_info, name)) File

131 "/anaconda2/lib/python2.7/site-packages/pip/\_vendor/pkg\_resources/ init .py",

132 line 1823, in \_has return zip\_path in self.zipinfo or zip\_path in self.\_index() File

133 "/anaconda2/lib/python2.7/site-packages/pip/\_vendor/pkg\_resources/ init .py",

134 line 1703, in zipinfo return self.\_zip\_manifests.load(self.loader.archive) File

135 "/anaconda2/lib/python2.7/site-packages/pip/\_vendor/pkg\_resources/ init .py",

136 line 1643, in load mtime = os.stat(path).st\_mtime OSError: [Errno 2] No such file or

137 directory: ’/anaconda2/lib/python2.7/site-packages/PyPrind-2.11.1-py2.7.egg’

138 10. Your ipynb files have some simple deprecation warnings.

139 11. In the ipynb guides, you should use "predictions[predictions.predicted == 1].head()" to

140 display results of predictions so that people can examine the potentially matching entities to

141 understand that the matching has actually occurred.

142 12. In the ipynb files, the notebooks say that the RF model has highest precision and recall, but

143 I see the linear regression has highest precision and recall.

144 13. Giving an idea of the computational runtime of the debugger on some data set would be

145 nice, or at least having a progress bar (like with the regular blocker) would be nice so that

146 a user can gauge how long it will take to run the debugging phase. (I ire running it

147 initially with 10,956,134 tuple pairs)

148 14. The AttrEqBlocker doesn’t seem to work for us on our tables when matching equivalence of

149 years.

150 15. It’s not clear to me how to load a list of labels from a file for the labeling portion. Manually

151 labeling all of the data within the .ipynb isn’t very practical for larger data sets, and it’s not

152 clear to me how to add labels to the data set in a manner that also generates ther required

153 metadata.