Report

You must also submit (on your project page) a pdf file that describes the following:

• For each of the five learning methods (Decision Tree, Random Forest, SVM, Naive Bayes, Logistic Regression), report the precision, recall, and F-1 that you obtain when you perform cross validation for the first time for these methods on I.

	Precision	Recall	F1
DecisionTree	0.883431	0.965278	0.920400
RandomForest	0.883431	0.965278	0.920400
SVM	0.866166	0.965278	0.910821
NaiveBayes	0.702381	0.976389	0.814279
LogReg	0.882955	0.955278	0.915278

• Report which learning based matcher you selected after that cross validation.

After the first cross validation Decision Tree and Random Forest both had the same precision, recall and F1 scores of 0.883431, 0.965278, 0.920400 respectively so either of these work as the best matcher. We selected Decision Tree because we have to pick one and Decision Trees are conceptually simpler.

For each cross validation iteration, report (a) what matchers were you trying to evaluate using the cross validation, and (b) precision/recall/F-1 of those.

For each iteration we were evaluating Decision Tree, Random Forest, SVM, Naive Bayes, Logistic Regression

Iteration 1: CV

	Precision	Recall	F1
DecisionTree	0.883431	0.965278	0.920400
RandomForest	0.883431	0.965278	0.920400
SVM	0.866166	0.965278	0.910821
NaiveBayes	0.702381	0.976389	0.814279

LogReg	0.882955	0.955278	0.915278

• Report all debugging iterations and cross validation iterations that you performed. For each debugging iteration, report (a) what is the matcher that yoeu are trying to debug, and its prcision/recall/F-1, (b) what kind of problems you found, and what you did to fix them, (c) the final precision/recall/F-1 that you reached.

Iteration 2: Debug

Upon our debug iteration using Decision trees, we achieved a precision, recall and F1 of 72, 88, and 80% respectively using a sample train and test. We noticed all false positive and false negative cases involved slight variations with names. In most cases, games shared the same gameplay_time and num_players, but the names indicated different editions, variations, and types (e.g. expansions). Some false positive instances the “Killer Bunnies and the Quest for the Magic Carrot RED Booster” and “Killer Bunnies and the Quest for the Magic Carrot VIOLET Booster”, which were the same game but with different skins (and thus not a match in our golden label) but sharing the same gameplay attributes. Or “Air Superiority” and “Air Strike”, which were the main game and expansion to the game respectively, thus sharing the same gameplay attributes. On the false negative end, the 2 games (6 nimmt! and Tally Ho!) each had discrepancies between the gameplay attribute values, despite being the same games in both tables.

We considered adding a name generated feature via a edit distance score on each table’s game names. Using this new feature, we saw an increase in precision, recall, and F1 to 85, 92, 88% across all three respectively. Among false positives, we particularly noticed games which had similar names, but we in fact not the same game. “Killer Bunnies and the Quest for the Magic Carrot RED Booster” and “Killer Bunnies and the Quest for the Magic Carrot VIOLET Booster” came back as a false positive match, particularly notable for their very similar names despite a difference in one word. On the other side, all the games which were false negative were games whose names varied in the inclusion of edition number or not, such as “1815: The Waterloo Campaign” and “1815 - The Waterloo Campaign (1st Edition)”, which referred to the same game but had a high edit distance between these 2 name strings due to extra identifiers such as editions.

Iteration 3: CV

	Precision	**Recall**	**F1**
DecisionTree	**0.882392**	**0.920359**	**0.898782**
RandomForest	**0.910866**	**0.931013**	**0.919163**

SVM	**0.937895**	**0.898137**	**0.916329**
NaiveBayes	**0.821319**	**0.964624**	**0.882312**
LogReg	**0.942857**	**0.943513**	**0.941584**

• Report the final best learning-based matcher that you selected, and its precision/recall/F-1.

TODO
Based on the results of iteration 3 with cross validation, the classifier with relatively high and consistent metrics across precision, recall, and F1 was **LogReg,** with around 94% across all the aforementioned metrics.

It is important to note that all precision/recall/F-1 numbers asked for in the aboves are supposed to be numbers obtained via CV on the set I.

• Now report the following:
– For each of the five learning methods, train it on I, then report its precision/recall/F-1 on J.

DecisionTree
Precision : 94.29% (33/35)
Recall : 97.06% (33/34)
F1 : 95.65%
False positives : 2 (out of 35 positive predictions)
False negatives : 1 (out of 65 negative predictions)

RF
Precision : 91.67% (33/36)
Recall : 97.06% (33/34)
F1 : 94.29%
False positives : 3 (out of 36 positive predictions)
False negatives : 1 (out of 64 negative predictions)

NaiveBayes
Precision : 80.95% (34/42)
Recall : 100.0% (34/34)
F1 : 89.47%
False positives : 8 (out of 42 positive predictions)
False negatives : 0 (out of 58 negative predictions)

LogReg

Precision : 94.29% (33/35)
Recall : 97.06% (33/34)
F1 : 95.65%
False positives : 2 (out of 35 positive predictions)
False negatives : 1 (out of 65 negative predictions)

SVM
Precision : 97.06% (33/34)
Recall : 97.06% (33/34)
F1 : 97.06%
False positives : 1 (out of 34 positive predictions)
False negatives : 1 (out of 66 negative predictions)

– For the final best matcher Y∗, train it on I then report its precision/recall/F-1 on J
LogReg
Precision : 94.29% (33/35)
Recall : 97.06% (33/34)
F1 : 95.65%
False positives : 2 (out of 35 positive predictions)
False negatives : 1 (out of 65 negative predictions)

Surprisingly, LogReg did not do as well as SVM on the final train-test set accuracy. However, it did have relatively high metrics (94%+) across all three and still proves a very reliable classifier (based on these results)

Final set of features are:

- Edit distance on min gameplay time
- Edit distance on max gameplay time
- Edit distance on min recommended number of players
- Edit distance on max recommended number of players
- Edit distance on game name

• Report an approximate time estimate: (a) how much did it take to label the data, and (b) to find the best learning-based matcher.

To label the data, we spent 1.5 hours generating random samples from the data set and labeling them. During our first iteration of labeling, we noticed about less than 12% positive of the data had positive golden labels. In order to increase the positive rate from blocking, we extended the set of stop_words that should not be considered when running overlap blocking using Jaccard, particularly by adding “of”, “edition”, “game”, “expansion”. We then reran blocking, resampled from the new blocking candidate set, and relabeled. We achieved a 31.3% positive rate on data in this new set, which we used for the final matching.

It took us around 5 hours to find the best learning-based matcher.

Discuss why you can't reach higher precision, recall, F-1.

We can't get higher precision without sacrificing recall or vice versa because of the nature of how names can look very similar but be different and can look very different but be the same. For instance "Killer Bunnies and the Quest for the Magic Carrot RED Booster" and "Killer Bunnies and the Quest for the Magic Carrot VIOLET Booster" look similar, but the one different word is enough to assure us that they do in fact refer to different things. Our classifier gives these a match despite them not being the same thing because the names are so similar. We could try to make it more sensitive to differences in name but we already have problems with it not being sensitive enough in some situations. For example "Monopoly: NASCAR" and "Monopoly - Nascar Nextel Cup (Collector's Edition)" are in fact the same game (we checked) but our matcher classifies them as different because the names are so different. Thus adjusting the sensitivity to fix one problem would create more of the other type of problem. Ideally, we would embed some semantic measure of what these individual words are referring to in order to better match certain words but not other (e.g. edition number vs game naming). We can't really mitigate this problem any more by using other non-name based features because all of the ones that are on both sites are already in use and can't be made better because of differences in gameplay time and recommended ages between the two sites (in theory they should be the same because most game boxes come with that information printed on them but for some games it seems people just took their best guess).

Other Notes

During our project, we noticed some issues and bugs in the py_entitymatching code which we submitted issues to github regarding the OverlapBlocker and stop_words. Our submission and write up is available here:
https://github.com/anhaidgroup/py_entitymatching/issues/46
https://github.com/anhaidgroup/py_entitymatching/issues/44