CS839 –Project Stage 3 Report

Team Members:

- Sreyas krishna Natarajan(snatarajan6@wisc.edu)
- Sugadev Chellakkannu(chellakkannu@wisc.edu)
- Vignesh Thirunavukkarasu(thirunavukka@wisc.edu)

Entity of choice:

The entity that we chose was music albums, majorly from genres – pop and rock.

a) **Table A – BestBuy:** The table from BestBuy contains the music albums' title, artist, genre, release and rating**.**

b) **Table B – Metacritic:** The table from Metacritic contains the music albums' title, artist, genre, release and rating.

The sites apparently did not have the same schema and we had choose to select the above 5 columns: Title, artist, genre, release and rating; to get the same schema for both.

Tuples:

BestBuy – 4360
Metacritic – 4330

Schema:

ID - Unique identifier to identify the tuples
Title – name of the music album
Artist – artist of the album
Genre - conventional category that identifies some pieces of music
Release – date of release of the album
Rating – rating the album has received on that particular site.

Blocker Used:

We analysed our two tables and found that **overlap blocker** is the best choice (even we experimented with other blockers but got better results with the overlap blocker).

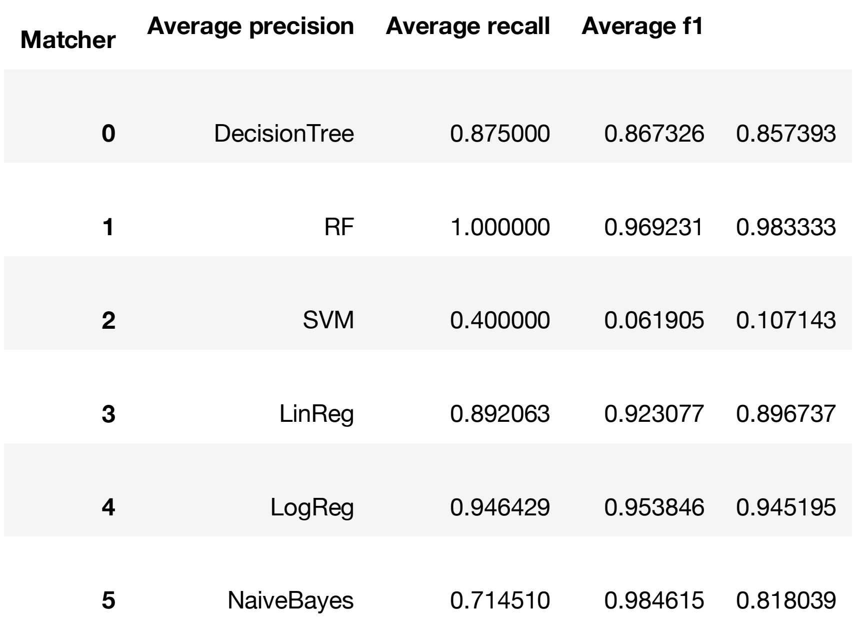
Overlap blocker:

The overlap blocker helped us find the overlapping strings as either words or q-grams. This helped us narrow down with overlaps between Title, Artist and Genres and removed the non-matching tuples from the dataset.

Number of tuple pairs in the candidate set obtained after the blocking step:
987

Number of tuple pairs in the sample G that you have labelled: 450

For each of the six learning methods provided in Magellan (Decision Tree, Random Forest, SVM, Naive Bayes, Logistic Regression, Linear 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.



Matcher	Average precision	Average recall	Average f1	
0	DecisionTree	0.875000	0.867326	0.857393
1	RF	1.000000	0.969231	0.983333
2	SVM	0.400000	0.061905	0.107143
3	LinReg	0.892063	0.923077	0.896737
4	LogReg	0.946429	0.953846	0.945195
5	NaiveBayes	0.714510	0.984615	0.818039

Learning based matcher you selected after that cross validation:
We selected Logistic Regression after the cross validation

Report all debugging iterations and cross validation iterations that you performed. For each debugging iteration, report (a) what is the matcher that you are trying to debug, and its precision/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. 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.
Precision, Recall, and F-1 for the six classifiers:

For each Debugging iteration,

a) We were trying to debug the **Decision tree** matcher and **Logistic regression matchers.** Because both of them seems to perform well for us.

Initially,
D Trees: Precision – 0.875, **Recall –** 0.867326, **F-1 –** 0.857393
Logistic Regression: Precision – 0.94629, **Recall –** 0.953846, **F-1 –** 0.945195

b) Problems found:

- When using the blockers to reduce the number of unwanted tuples. We experimented with overlap blockers of with overlap_size of 2 & 3.

- The problem arose when the titles and artist names which were only 1 word long were completely discarded because of the size selected resulting in lesser matches.

- We tried to use the Levenshtein similarity measure between the two input strings using a rule-based blocker. But the blocker scrutinized too much on the names and reduced the precision and recall of the matchers.

We fixed these problems by using an overlap blocker on three different attributes Title, Artist and Genre.

Title – overlap-size = 1 with overlap on words

Artist – overlap-size = 1 with overlap on words
Genre – overlap-size = 3 with q-grams of size 3. (Pop was the smallest q-gram)

This resulted in good precision and recall with the learning- based matchers.

c) **Final Precision, Recall, F-1 after debugging iterations:**

Decision Tree: Precision: 1.0000, **Recall:** 0.87667, **F1:** 0.9329
Logistic Regression: Precision: 1.0000, **Recall:** 0.87667, **F1:** 0.9329

For each Cross-validation iteration:

a) We were trying to evaluate Logistic Regression and Decision Trees with each cross validation

b) Precision, Recall, F-1 of the matchers:
Decision Tree: Precision: 1.0000, **Recall:** 0.87667, **F1:** 0.9329
Logistic Regression: Precision: 1.0000, **Recall:** 0.9667, **F1:** 0.9818

Final Learning based matcher, you selected after that cross validation:

We selected Decision Trees classifier after the cross-validation step because it showed consistent improvement for our usage.
Decision Tree: Precision: 1.0000, **Recall:** 0.87667, **F1:** 0.9329

Six Learning Methods on J (Test Set):

Decision Tree:
```
Precision : 95.0% (19/20)
Recall : 86.36% (19/22)
F1 : 90.48%
False positives : 1 (out of 20 positive predictions)
False negatives : 3 (out of 115 negative predictions)
```

Random Forest: `Precision : 100.0% (18/18)`
```
Recall : 81.82% (18/22)
F1 : 90.0%
```

```
False positives : 0 (out of 18 positive predictions)
False negatives : 4 (out of 117 negative predictions)
```

SVM: `Precision : 0.0% (0/0)`
```
Recall : 0.0% (0/22)
F1 : 0.0%
False positives : 0 (out of 0 positive predictions)
False negatives: 22 (out of 135 negative predictions)
```

Linear Regression:
```
Precision : 81.82% (18/22)
Recall : 81.82% (18/22)
F1 : 81.82%
False positives : 4 (out of 22 positive predictions)
False negatives : 4 (out of 113 negative predictions)
```

Logistic Regression:
```
Precision : 86.36% (19/22)
Recall : 86.36% (19/22)
F1 : 86.36%
False positives : 3 (out of 22 positive predictions)
False negatives : 3 (out of 113 negative predictions)
```

Naïve Bayes:
```
Precision : 65.52% (19/29)
Recall : 86.36% (19/22)
F1 : 74.51%
False positives : 10 (out of 29 positive predictions)
False negatives : 3 (out of 106 negative predictions)
```

Best Learning Based Matcher Y on Test Set J :

Decision Tree:
```
Precision : 95.0% (19/20)
Recall : 86.36% (19/22)
F1 : 90.48%
False positives : 1 (out of 20 positive predictions)
False negatives : 3 (out of 115 negative predictions)
```

Approximate Time estimates:

a) To do Blocking: 3 hours
b) To Label data: 2 hours
c) To Find the Best Matcher: 1 hour

WAY TO IMPROVE RECALL:

We feel that the recall of 86% is good enough for a classifier. However inorder to improve the recall further, we could try adding whitelist to the classifier and increase/alter the features by further analysing the false negatives.

COMMENTS ON MAGELLAN:

MERITS:

- Easy to learn and as a naïve programmer.
- The tool is highly developer friendly.
- Debugger is a very good feature that greatly reduces the developing time.

AREAS OF IMPROVEMENT AND FEEDBACK:

- We found few **duplicates** in the candidate set that survived the blocking step. This might be an issue with the Magellan or due to duplicate records in the input tables. It would be better if Magellan could incorporate some feature to show/remove the duplicate records. It makes sense on matching with the same table as discussed in class but when matching with a second table, it would be better to show/remove the duplicate records.

- We feel that with the overlap blocker we need to be able to consider one-word entries and not discard them when the overlap_size > 1. Artist names like 'Sia' and Titles like 'x' were completely discarded.

- Providing so many options for similarity measures with rule-based blockers proved very useful and helped us understand new similarity measures.

- Some errors generated by Magellan are not completely comprehensive.
  Eg.:
  ```
  /home/user/miniconda3/lib/python3.6/site-packages/sklearn/metrics/classification.py:1135: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 due to no predicted samples. 'precision', 'predicted', average, warn_for)
  ```

- It would be better if some video tutorial is provided for Magellan.

- We somehow feel that the guide for Magellan is not sufficient enough to give the complete understanding. We hope once the book on Magellan is released as the professor said in the class, it would be good.

- Currently, Magellan matches only two tables, it would be better if support for multiple tables is given. The only way to do now is match two tables and then use the matched results to match with the third table.