

Experiment Design for Data Science: Exercise 2

Paper Reproduction: Option 1

Konstantin Damjanovic
01151948

Moritz Leidinger
11722966

Dzan Operta
11935976

ABSTRACT

In this project we are trying to reproduce the results, mainly the three tables featured, of the paper *TUD-MMC at MediaEval 2016: Context of Experience task* [1], which is on the other using a data set described by the paper *Right in flight? A dataset for exploring the automatic prediction of movies suitable for a watching situation. In Proceedings of the 7th International Conference on Multimedia Systems* [2]

1 DATA PREPROCESSING

The paper worked with different modalities:

- Audio data
- Visual data
- Metadata
- User ratings

The training and test data split was already done, as the corresponding files were in separate folders. However, finding the corresponding target values, the binary variable if the movie is good to watch on an airplane or not, was not as straight forward. While the spreadsheet that listed all movies that were in the dev set, contained the column 'goodforairplane', the file for the test set did not. In the whole folder structure there were some spreadsheets labeled 'test set' that contained the target value, but none of these had a complete match with the data files in the folders. We ended up merging the test data file with the 'dataset_complete.xlsx', in the tab 'test', from the mediaeval folder. This leaves us with 95 movies in the training set and 223 movies in the test set, for a total of 318. The detail on the steps on data processing that was done really varied, so we will be discussing the different types of data separately in more detail.

1.1 Audio Data

The audio data was comprised of one file per movie, each having 13 rows. The number of columns changed from movie to movie, but was constant for each file. We assumed the length was some kind of measure per frame or other increment of the triler. This one was probably the most well described part of the data sets in the paper, so the preprocessing was pretty straightforward. They mention each row represents a Mel-frequency cepstral coefficient (MFCC) and that they took the average per row, filling any missing value with 0. We did just that and formed 13 audio features per movie.

1.2 Visual Data

The visual data was not as transparent. Again provided was one file per movie, but this time with a constant dimension, 2 rows and 816 columns. The paper mentioned these being JCD, or Joint Composit Descriptors, which are used to describe and compare images. However, despite reading the corresponding papers, it was unclear how this related to the given values. They varied vastly per row,

and the overall distributions per row were completely different between movies. The only consistency was that per column values were pretty consistent, so we had two approaches to deal with this part of the data set: Choosing all entries as or taking the average per column. This left us with 1652 or 826 visual features.

1.3 Textual Data

In this case we had one file, one for the test and one for the training set. From the file name, 'tf_idf', we deduced that these spreadsheets were term frequency-inverse document frequency matrices, which describe the importance of keywords in text. There were thousands of different keywords in the headers, varying in number between the two sets. The values stopped after the 95th and the 223rd column, making it quite obvious that the matrix was supposed to be transposed and the movies added as row names. We added the movies in alphabetical order, since that made the most sense. Since we did not know what exactly the values meant, we turned each entry into a dummy variable, so it showed whether a keyword was important or not. At last we removed any keyword from the test data that was not contained in the training data and inserted any missing keyword with zeros for all movies. This left us with 3284 textual features.

1.4 Metadata

1.5 User Score

2 REPRODUCING TABLE 1

The first table to reproduce used a rule-based PART classifier in WEKA to evaluate Precision, Recall and F1 Score for User ratings, Visual, Metadata, Metadata + user rating and Metadata + visual. The result of the paper and ours in comparison can be seen in the following two tables. This was quite clearly outlined in the paper, however, the results were exactly the same as the first table in the *Right in flight?*-paper, which made us suspect that they just copying the table as a whole. Our solutions vary a bit from the table in the paper, especially user ratings, which had basically no result for precision and F1.

3 REPRODUCING TABLE 2

The second table we had to reproduce required two methods

3.1 Las Vegas Wrapper

3.2 scikit part 2

Modifying the template — including but not limited to: adjusting margins, typeface sizes, line spacing, paragraph and list definitions, and the use of the \vspace command to manually adjust the vertical spacing between elements of your work — is not allowed.

Your document will be returned to you for revision if modifications are discovered.

4 REPRODUCING TABLE 3

5 TITLE INFORMATION

The title of your work should use capital letters appropriately - <https://capitalizemytitle.com/> has useful rules for capitalization. Use the `title` command to define the title of your work. If your work has a subtitle, define it with the `subtitle` command. Do not insert line breaks in your title.

If your title is lengthy, you must define a short version to be used in the page headers, to prevent overlapping text. The `title` command has a “short title” parameter:

```
\title[short title]{full title}
```

6 AUTHORS AND AFFILIATIONS

Each author must be defined separately for accurate metadata identification. Multiple authors may share one affiliation. Authors’ names should not be abbreviated; use full first names wherever possible. Include authors’ e-mail addresses whenever possible.

The `authornote` and `authornotemark` commands allow a note to apply to multiple authors — for example, if the first two authors of an article contributed equally to the work.

If your author list is lengthy, you must define a shortened version of the list of authors to be used in the page headers, to prevent overlapping text. The following command should be placed just after the last `\author{}` definition:

```
\renewcommand{\shortauthors}{McCartney, et al.}
```

Omitting this command will force the use of a concatenated list of all of the authors’ names, which may result in overlapping text in the page headers.

REFERENCES

- [1] Cynthia C. S Liem and Bo Wang. *TUD-MMC at MediaEval 2016: Context of Experience task*, 2016.
- [2] M. Riegler, M. Larson, C. Spampinato, P. Halvorsen, M. Lux, J. Markussen, K. Pogorelov, C. Griwodz, and H. Stensland. *Right in flight? A dataset for exploring the automatic prediction of movies suitable for a watching situation*. In *Proceedings of the 7th International Conference on Multimedia Systems*, pages 45:1–45:6. ACM, 2016.

A TABLES

Table 1 from the paper

Features used	Precision	Recall	F1
User rating	0.371	0.609	0.461
Visual	0.447	0.476	0.458
Metadata	0.524	0.516	0.519
Metadata + user rating	0.581	0.6	0.583
Metadata + visual	0.584	0.6	0.586

Table 2 from the paper

Classifier	Modality	Precision	Recall	F1
K-Nearest neighbor	metadata	0.607	0.654	0.63
Nearest mean classifier	metadata	0.603	0.579	0.591
Decision tree	metadata	0.538	0.591	0.563
Logistic regression	metadata	0.548	0.609	0.578
SVM (Gaussian Kernel)	metadata	0.501	0.672	0.574
Bagging	metadata	0.604	0.662	0.631
Random Forest	metadata	0.559	0.593	0.576
AdaBoost	metadata	0.511	0.563	0.536
Gradient Boosting Tree	metadata	0.544	0.596	0.569
Naive Bayes	textual	0.545	0.987	0.702
K-Nearest neighbor	textual	0.549	0.844	0.666
SVM (Gaussian Kernel)	textual	0.547	1	0.707
K-Nearest neighbor	visual	0.582	0.636	0.608
Decision tree	visual	0.521	0.55	0.535
Logistic regression	visual	0.616	0.6	0.608
SVM (Gaussian Kernel)	visual	0.511	0.67	0.58
Random Forest	visual	0.614	0.664	0.638
AdaBoost	visual	0.601	0.717	0.654
Gradient Boosting Tree	visual	0.561	0.616	0.587
Logistic regression	audio	0.507	0.597	0.546
Gradient Boosting Tree	audio	0.56	0.617	0.58

Table 3 from the Paper

Stacking Strategy	Precision	Recall	F1
Voting (cv)	0.94	0.57	0.71
Label Stacking (cv)	0.72	0.86	0.78
Label Attribute Stacking (cv)	0.71	0.79	0.75
Voting (test)	0.62	0.8	0.7
Label Stacking (test)	0.62	0.9	0.73

Table 1 reproduction

Features used	Precision	Recall	F1
User rating	very low	0.610	very low
Visual	0.545	0.520	0.526
Metadata	0.462	0.448	0.454
Metadata + user rating	0.478	0.462	0.468
Metadata + visual	0.528	0.511	0.517

Table 2 reproduction

Classifier	Modality	Precision	Recall	F1	BF ^I
k-Nearest neighbor	audio	0.506	0.617	0.55	10
Decision tree	audio	0.62	0.657	0.622	4
SVM (Gaussian Kernel)	audio	0.51	0.737	0.543	13
Random Forest	audio	0.526	0.583	0.545	13
AdaBoost	audio	0.602	0.6	0.591	4
Gradient Boosting Tree	audio	0.595	0.657	0.619	6
k-Nearest neighbor	textual	0.523	0.77	0.619	1543
Decision tree	textual	0.547	0.823	0.653	116
Logistic regression	textual	0.533	0.617	0.558	3282
SVM (Gaussian Kernel)	textual	0.527	0.903	0.664	3282
Bagging	textual	0.615	0.753	0.658	3282
Random Forest	textual	0.559	0.82	0.654	3282
AdaBoost	textual	0.677	0.887	0.756	2375
Gradient Boosting Tree	textual	0.669	0.863	0.744	1007
Naive Bayes	textual	0.591	0.73	0.645	3211
k-Nearest neighbor	visual	0.614	0.677	0.641	757
Decision tree	visual	0.6	0.553	0.555	16
Logistic regression	visual	0.617	0.697	0.633	826
SVM (Gaussian Kernel)	visual	0.577	0.92	0.708	568
Bagging	visual	0.677	0.58	0.614	826
Random Forest	visual	0.561	0.607	0.563	198
AdaBoost	visual	0.606	0.6	0.592	826
Gradient Boosting Tree	visual	0.605	0.617	0.602	826
Naive Bayes	visual	0.548	0.687	0.588	826
k-Nearest neighbor	metadata	0.579	0.557	0.56	75
Decision tree	metadata	0.558	0.517	0.533	39
Logistic regression	metadata	0.569	0.54	0.536	22
SVM (Gaussian Kernel)	metadata	0.548	1.0	0.707	75
Random Forest	metadata	0.527	0.553	0.526	75
AdaBoost	metadata	0.664	0.54	0.576	75
Gradient Boosting Tree	metadata	0.541	0.567	0.548	75

Table 3 reproduction UPDATE!!!!

Stacking Strategy	Precision	Recall	F1
Voting (cv)	0.94	0.57	0.71
Label Stacking (cv)	0.72	0.86	0.78
Label Attribute Stacking (cv)	0.71	0.79	0.75
Voting (test)	0.62	0.8	0.7
Label Stacking (test)	0.62	0.9	0.73