Exercise 2: Reproduce experimental results from a paper

Option 1: Predicting the Suitability of Movies for an Inflight Viewing Context

Experiment Design for Data Science 2019W

Sebastian Dolezel
01126006
e11260061@student.tuwien.ac.at

Elisabeth Harlander
01106899
e1106899@student.tuwien.ac.at

Pia Pachinger
01408624
e14086241@student.tuwien.ac.at

ABSTRACT

This report marks our attempt to recreate the results from the **TUD-MMC at MediaEval 2016: Context of Experience**[2] paper. In the efforts to obtain results as similar as possible, we focus on the experimental design and reproducability of the template. Whenever the processing steps are not clearly defined we try to apply an approach that is reasonable in respect to the situation.

1 RULE-BASED PART CLASSIFIER: RECREACTION OF TABLE 1

The Results from table 1 are seen as initial experiments to determine the usefulness of the various features. The table compares the metrics from the paper [1] with our recreation attempt.

Difficulties. The paper [1] only mentioned the application of a rule-base PART classifier on the dataset, further information was not provided.

Strategies. In order to recreate the metrics we turned to the dataset paper [2] for more details. The authors used the WEKA machine learning library to calculate the weighted average of precision, recall and F1-score.
However, we did not know which version of WEKA the authors used, so we downloaded the latest version (WEKA 3.8.4).
We actually performed this task after the rebuilding of table 2 and used the preprocessed data from this approach (e.g. we used the visual data where we kept only the first row). Reading the files into WEKA was rather straightforward but the test options for the rule-based PART classifier were not declared in one of the papers either. So we decided to use the test sets of the features and let WEKA compute the scores. A resulting model is shown in figure 1.

Key Findings. It is not sufficient to state which classifier was applied, but also which software (-version) was used and which preprocessing steps were performed. WEKA gives the user a variety of options which makes it quite important to recite the performed steps to assure reproducability.
Nevertheless, we obtained satsifying results that were comparable to the ones in the papers.

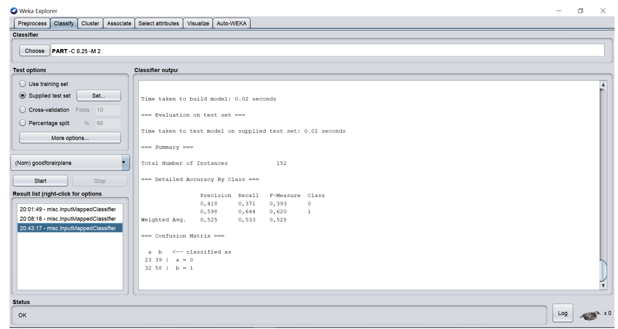

Figure 1: Screenshot of the resulting model for the metadata after applying the rule-base PART classifier

2 BASE CLASSIFIERS: RECREATION OF TABLE 2

Difficulties and Strategies. We could not find out, which Python and Scikit-Learn version was used. We simply used recent versions of both, because guessing, which versions the researchers had on their computers seemed hopeless to us, since they probably did not have the most recent versions installed. This problem is also crucial, because they used the default parameters of the Machine Learning models, which probably changed over the years.

Again, it was not depicted clearly, which data sets were used for computing table 2. We used the folders in the folder "Dev Set" of the folder "CoE dataSet".

10-fold cross validation leads to small test sizes applied to 95 data points. This is the size of the training data. Still, we imitated this approach. No random seed was provided for cross validation, we chose any seed.

The movie names in the different files were written slightly variably (apostrophes appeared for example in some files, in others not). It was extra work to merge this data.

No implementation of the Las Vegas Wrapper was specified in the paper. We implemented it ourselves, trying to imitate the Las Vegas Wrapper (optimizing the F1 score) described in the cited paper. We could not find out, how many different combinations of features were tried out. We used about 10 otherwise our results would have exceeded the original results notably. As described in the paper, we have filtered out the classifiers for all modalities, for which our predictions achieved a score of $F1 > 0,5$, which clearly indicates that either the default parameters or some mathematical functions have changed within the past updates of the used libraries, as we received a lot more results above the chosen baseline of random guessing (0.5).

3 CLASSIFIER STACKING: RECREATION OF TABLE 3

Difficulties and Strategies. It was not clear, whether the scores computed with cross-validation were computed on the training or the test data. We applied cross-validation only to the training data. For the training data, we used the data of the same folders as for table 2.

For computing the scores for the test data, we used the training data mentioned above for training the models (when applicable). We evaluated on the test data from the "Test Set" folder which can be found in the "CoE Dataset" folder.

[0] https://www.cs.waikato.ac.nz/ml/weka/ (accessed: 30.01.2020)

EDDS 2019W, , Vienna University of Technology

Features Used	Source	Precision	Recall	F1
User Rating	Paper	0.371	0.609	0.461
User Rating	Recreation	1.000	1.000	1.000
Visual	Paper	0.447	0.476	0.458
Visual	Recreation	0.493	0.503	0.489
Metadata	Paper	0.524	0.516	0.519
Metadata	Recreation	0.525	0.533	0.528
Metadata + User Rating	Paper	0.581	0.600	0.583
Metadata + User Rating	Recreation	0.528	0.520	0.523
Metadata + Visual	Paper	0.584	0.600	0.586
Metadata + Visual	Recreation	0.471	0.470	0.479

Table 1: Comparison of Table 1 in the reference paper and our recreation attempt using Weka

We did not include the audio and text data into our computations due the serious merging problems. These arose from the already mentioned differences in the namings of the movies.

It was not clear which classifier was used for the Label Stacking and the Label Feature Stacking. We used Logistic Regression as this is the default classifier used by the Ensemble Voting classifier from Sklearn. The Label Feature Stacking was not explained at all, we decided to concat the predictions of the base classifiers with the data itself apply a classifier to the resulting dataframe.

REFERENCES

[1] Michael Riegler, Martha Larson, Concetto Spampinato, Pål Halvorsen, Mathias Lux, Jonas Markussen, Konstantin Pogorelov, Carsten Griwodz, and Håkon Stensland. 2016. Right Inflight? A Dataset for Exploring the Automatic Prediction of Movies Suitable for a Watching Situation.

[2] Bo Wang and Cynthia C. S. Liem. 2016. TUD-MMC at MediaEval 2016: Context of Experience Task. *MediaEval 2016 Workshop* (2016).

Algorithm	Source	Precision	Recall	F1	Modality
Gradient Boosting Tree	Paper	0.560000	0.617000	0.587000	Audio
Logistic Regression	Paper	0.507000	0.597000	0.546000	Audio
AdaBoost	Reproduced	0.561667	0.566667	0.558889	Audio
Gradient Boosting Tree	Reproduced	0.552857	0.566667	0.555253	Audio
Logistic Regression	Reproduced	0.511429	0.563333	0.529297	Audio
SVM	Reproduced	0.467619	0.600000	0.522145	Audio
adaboost	Reproduced	0.725833	0.633333	0.646378	Metadata
bagging	Reproduced	0.592857	0.593333	0.580730	Metadata
decision_tree	Reproduced	0.595595	0.616667	0.593802	Metadata
gradient_boost	Reproduced	0.709167	0.673333	0.647529	Metadata
knn	Reproduced	0.666310	0.666667	0.639066	Metadata
logistic_regression	Reproduced	0.596032	0.730000	0.652597	Metadata
nearest_mean	Reproduced	0.623690	0.610000	0.601805	Metadata
random_forest	Reproduced	0.657897	0.596667	0.590458	Metadata
svm	Reproduced	0.547980	1.000000	0.707843	Metadata
knn	Paper	0.607	0.654	0.630	Metadata
nearest mean classifier	Paper	0.603	0.579	0.591	Metadata
decision tree	Paper	0.538	0.591	0.563	Metadata
logistic regression	Paper	0.548	0.609	0.578	Metadata
svm	Paper	0.501	0.672	0.574	Metadata
bagging	Paper	0.604	0.662	0.631	Metadata
random forest	Paper	0.559	0.593	0.576	Metadata
adaboost	Paper	0.511	0.563	0.536	Metadata
gradient boosting tree	Paper	0.544	0.596	0.569	Metadata
Naive Bayes	Paper	0.545000	0.987000	0.702000	Textual
SVM	Paper	0.547000	1.000000	0.700000	Textual
k-Nearest neighbor	Paper	0.549000	0.844000	0.666000	Textual
AdaBoost	Reproduced	0.505714	0.673333	0.573907	Textual
Bagging	Reproduced	0.538413	0.756667	0.618528	Textual
Decision tree	Reproduced	0.561865	0.773333	0.644376	Textual
Gradient Boosting Tree	Reproduced	0.563056	0.863333	0.675992	Textual
Logistic Regression	Reproduced	0.547980	1.000000	0.707843	Textual
Naive bayes	Reproduced	0.524405	0.630000	0.568470	Textual
Random forest	Reproduced	0.546508	0.693333	0.603247	Textual
SVM	Reproduced	0.547980	1.000000	0.707843	Textual
AdaBoost	Paper	0.601000	0.717000	0.654000	Visuals
Decision Tree	Paper	0.521000	0.550000	0.535000	Visuals
Gradient Boosting Tree	Paper	0.561000	0.616000	0.587000	Visuals
KNN	Paper	0.582000	0.636000	0.608000	Visuals
Logistic Regression	Paper	0.616000	0.600000	0.608000	Visuals
Random Forest (not stable)	Paper	0.614000	0.664000	0.638000	Visuals
SVM	Paper	0.511000	0.670000	0.580000	Visuals
AdaBoost	Reproduced	0.603095	0.700000	0.639340	Visuals
Decision Tree	Reproduced	0.673611	0.760000	0.690188	Visuals
Gradient Boosting Tree	Reproduced	0.648373	0.780000	0.705604	Visuals
KNN	Reproduced	0.569960	0.740000	0.638352	Visuals
Logistic Regression	Reproduced	0.591349	0.860000	0.696097	Visuals
Random Forest (not stable)	Reproduced	0.587540	0.700000	0.607749	Visuals
SVM	Reproduced	0.536310	0.920000	0.673959	Visuals

Table 2: Comparison of Table 2 in the reference paper and our recreation attempt

Stacking Strategy	Source	Precision	Recall	F1
Voting (CV)	Paper	0.94	0.57	0.71
Voting (Train)	Recreation	0.59	0.82	0.68
Label Stacking (CV)	Paper	0.72	0.86	0.78
Label Stacking (CV)	Recreation	0.64	0.67	0.64
Label Attribute Stacking (CV)	Paper	0.71	0.79	0.75
Label Attribute Stacking (CV)	Recreation	0.58	0.79	0.66
Voting (Test)	Paper	0.62	0.80	0.70
Voting (Test)	Recreation	0.58	0.72	0.64
Label Stacking (Test)	Paper	0.62	0.90	0.73
Label Stacking (Test)	Recreation	0.53	0.36	0.43

Table 3: Comparison of Table 3 (Classifier Stacking Results) in the reference paper and our recreation attempt