Student Number: 236636

1. Introduction

CNN and GIST features were collected on pictures that were hand-labelled as memorable or not. We implemented a Random Forest Classifier (RFC) [1] to predict their label. We experimented with the hyper-parameters of the RFC and with feature pre-processing techniques, but found they had no positive effect on our model's accuracy. Our final model reached an accuracy of 75% on the validation set. We found that the subjective nature of memorability and the similarity in collected features that two pictures in different classes can have makes the data impractical to classify.

2. Approach

Our method creates many classifiers and aggregates their decision to make a prediction. It uses randomised subsets of the training data to make many decision trees, preventing over-fitting. Each tree finds the features that best split its subset of samples, essentially performing feature selection [2]. To predict, each tree makes a prediction on the same sample, and the most popular prediction is outputted. Therefore, the only assumption RFC makes is that the training samples are representative of the entire data set [3]. These advantages of RFC makes it a good theoretical solution for our classification problem.

3. Methods

3.1. Training

Our training data is split in half to create training and validation subsets. To train a classifier, we fit the training samples and their ground truth labels to the model.

3.2. Testing

We evaluate the performance of our classifier by repeating the steps from the data splitting, fitting and evaluating the model to measure its accuracy multiple times and averaging it out (see algorithm 1).

3.3. Model selection

We chose our model through theory, by looking at the data's characteristics and the nature of the task. The dataset has 4096 features for 3400 samples: high dimensionality and few samples. This creates the need for a classifier that can sort through lots of features and avoid over-fitting. We found that RFC fits those criteria [2]. We also implemented other models (SVC, K-means, Gradient Boosting) and found RFC outperformed them all.

Algorithm 1 Evaluate a Random Forest Classifier

Input: Samples X and their labels Y
Output: Average metric of RFC fitted on X and Y

```
metricTotal = 0
for iteration = 1, 2, ..., 10 do
    TrainData, ValData = splitData(X,Y)
    rfc = initialise and fit RFC to TrainData
    predictions = GetPredictions(ValData)
    metricTotal = GetMetric(predictions,ValData)
end for
averageMetric = metricTotal/10
return averageMetric
```

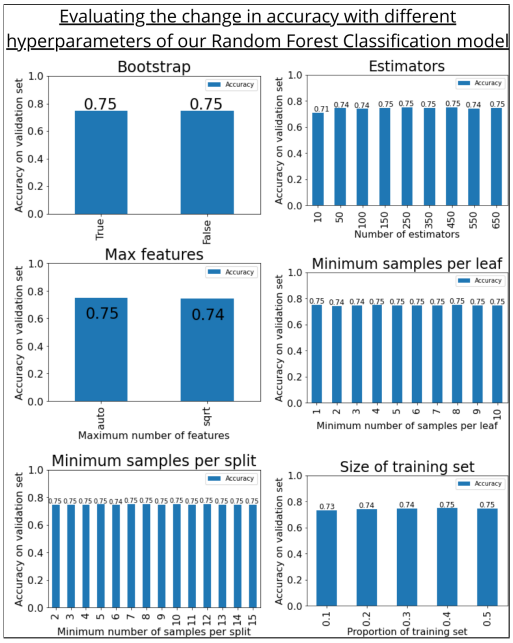

Figure 1. We experimented with 6 different hyper-parameters of our RFC model and plotted their average accuracy. This provides the experimental evidence to choose the hyper-parameters of our model that will maximise its performance.

3.4. Hyper-parameter selection

We evaluated the change in accuracy of RFC over 6 different hyper-parameters (Fig. 1) in order to find the ones that maximises it. They are the same ones we used throughout the evaluation of the pre-processing methods.

3.5. Data re-scaling

RFC performs feature selection by choosing the features that best split the data. Therefore, re-scaling it did not have any impact. We trained a model on unscaled data, normalised data and standardised data and found the accuracy

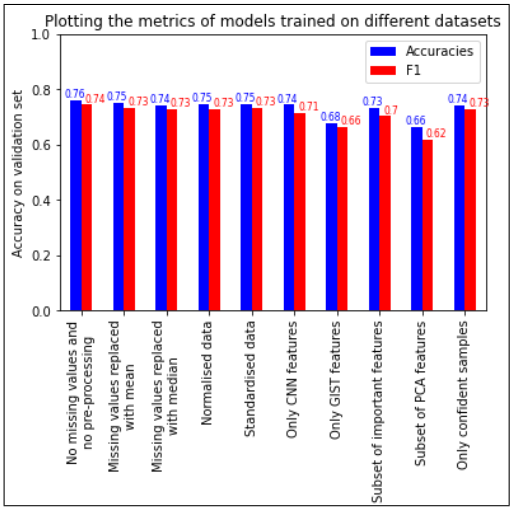

Figure 2. Metrical comparison of evaluated RFC models with the same hyper-parameters, trained and validated on different data sets. The impact of the training set is measured as the average accuracy- and F1-score on the validation set over 10 runs.

was not impacted (Fig. 2).

3.6. Feature selection

RFC builds decision trees according to the feature that best splits the data, performing feature selection on its own. We prove this by implementing pre-processing feature selection methods: only CNN features, only GIST features, PCA features (experimented to find the best subset of them), and important features (same experimentation as PCA). We found that they never outperformed the performance of the model trained on all features (Fig. 2).

3.7. Sample selection

Our samples have a confidence score for their label. We filtered out the 'not confident' samples and trained an RFC on the 'confident' ones. It performed worse than an RFC trained on all samples (Fig. 2).

3.8. Inputting data

We imputed the missing data with the mean and the median and evaluated two models trained on each to see which gave us the best performance. Imputing with the mean gave us the best accuracy, even if only by a slight margin (Fig. 2). The model trained on data with no missing values was unreliable: 68% accuracy on mean imputed data.

4. Results and discussion

4.1. Results

Our final model's accuracy is 75%. It is specific as it predicts more pictures as non-memorable (Fig. 3) even if

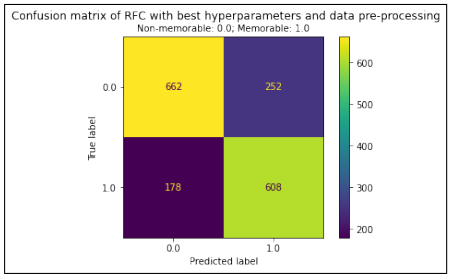

Figure 3. Confusion matrix of our final model, with the only pre-processing being the imputing of missing values with the mean. The hyper-parameters were chosen through the experimentation results in figure . 1.

the data-set is balanced. Furthermore, it performs 1% better on the confident data than on the un-confident.

4.2. Discussion

4.2.1 Improving performance

Improving the performance would require a more extensive grid search of every hyper-parameter of the RFC, and more data. We could have also tried different imputing methods, such as missForest [4]. Furthermore, our model could improve if the data was less subjective and had more labellers.

4.2.2 Improving evaluation

We could have pre-defined validation sets, with different imputing methods applied, to standardise our model evaluations. We could have also crossed the training and validation sets, such as training on mean-imputed data and validating on median-imputed data.

4.2.3 Subjective data-set

We found that 74% of the samples in the training data are unconfident, revealing the data's subjectivity. The labellers' lack of agreement would make any classifier struggle. Many pictures can have very similar features, but be classified differently. Even when excluding the unconfident data from the training set, the classifier did not perform better 2. Essentially, what makes a picture memorable or not for humans is not a feature that was collected by these CNNs and GISTs.

5. Results and discussion

In conclusion, our model was able to accurately predict if a picture is memorable from CNN and GIST collected features 3/4 of the time. We found a Random Forest Classifier almost worked out-of-the-box. We learned that a classification model's performance is limited by the confidence of its ground truth labelling.

References

[1] L. Breiman. *Machine Learning*, 45(1):5–32, 2001. 1

[2] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka. Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, 7(1), July 2020. 1

[3] D. L. (https://stats.stackexchange.com/users/11639/dmitry laptev). Random forest assumptions. Cross Validated. URL:https://stats.stackexchange.com/q/59182 (version: 2013-05-16). 1

[4] D. J. Stekhoven and P. Bühlmann. MissForest—nonparametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 10 2011. 2