Can Humour Styles be used to predict age?

Module Name: COMP2261
Date: 24th January 2022
Submitted as part of the degree of MSci in Computer Science and Mathematical Sciences within the Natural Sciences programme to the
Board of Examiners in the Department of Computer Sciences, Durham University

Abstract—This report documents the process of creating 3 machine learning models to predict age from data about peoples' humour styles. If the models are successful and accurate, it will mean there is significant evidence to suggest that humour styles vary with age.

Index Terms—Machine Learning, Python, Regression, Scikit-learn

1 Introduction

In this project I will carry out a machine learning task in order to train 3 different models to predict a person's age given a set of responses from a Humour Styles Questionnaire. I will use data collected from a 2003 study by Martin, Puhlik-Doris, Larsen, Gray and Weir [1] consisting of peoples' responses to 32 questions about the style of jokes they make or enjoy. Through carrying out this project I will find out how much peoples' senses of humour vary with age (if at all), by evaluating 3 machine learning models' ability to predict ages from the data-set.

2 Methodology

In order to create age predicting models I programmed using Python with the libraries: scikit-learn [2], NumPy [3] and Matplotlib [4].

2.1 Data Overview

I began this project by carrying out some preliminary data analysis on the data-set. The Humour Styles Questionnaire (HSQ) consisted of 32 questions, in which users rate how often a statement applies to them on a scale of 1 to 5 (if no answer is selected -1 is recorded in the data-set). These responses were then used to calculate four scores representing particular characteristics of the respondent's humour style. Also recorded in the data was the respondent's age and gender, as well as how accurate they thought their answers were on a scale from 0 to 100. The data-set used for this project contains a sample of 1072 responses of which 581 of the respondents were male, 477 were female, and 13 did not identify as either. As I don't know the distribution of the population overall for this data-set, I have to assume that the data sample i have access to is unbiased aside from age.

The maximum age in the sample was technically 44849, however given that the oldest person to ever live was 122 [5], this response along with 3 others should be disregarded. After removing all the clearly incorrect age samples, the actual maximum age was 70. The average respondent age was 26, and the minimum was 14. The median age was 22, and as shown in Figure 1 and Figure 2 below, the ages are heavily positively skewed - 75% of respondents were aged 31 or younger:

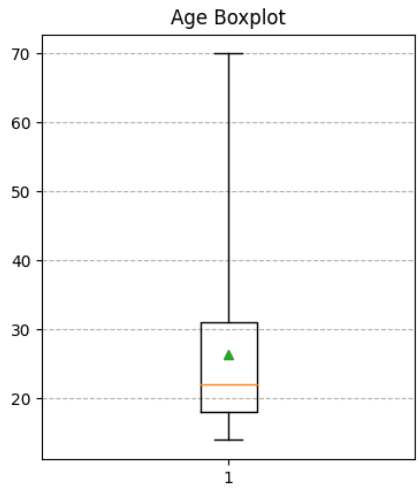

Fig. 1. Age Boxplot - The age data is heavily positively skewed

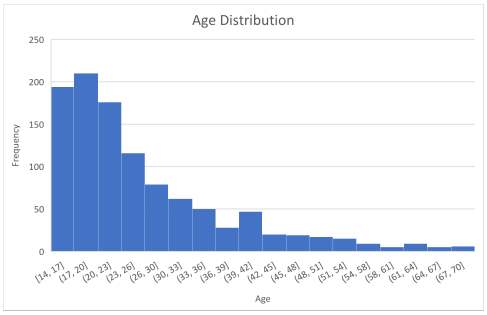

Fig. 2. Age Histogram

Age is largely independent from gender within the data-set - Figure 3 shows similar age distributions for each gender identification:

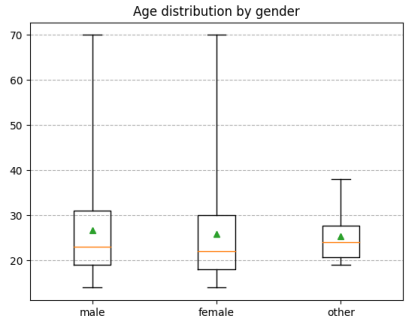

Fig. 3. Age boxplots by gender

Lastly, as shown in Figures 4, 5, 6 and 7, there is no immediately discernible correlation between age and any of the 4 calculated HSQ scores:

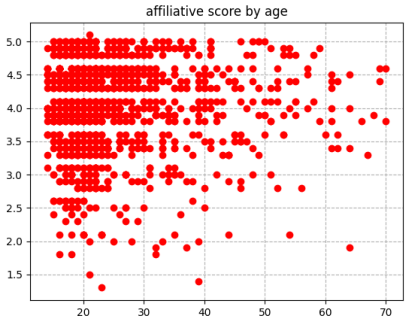

Fig. 4. Affiliative score plotted against Age

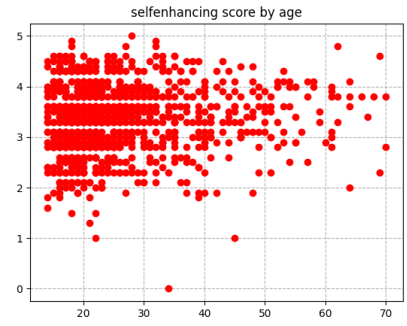

Fig. 5. Self-enhancing score plotted against Age

2.2 Data Preparation

Next I began preparing the data for use in machine learning algorithms. I first had to carry out data cleaning - this entailed ensuring that no incomplete or incorrect data was used in training the models. First I removed all instances with invalid ages (greater than 124 to be safe - there were no instances with age between 70 and 151 so the exact cutoff number wasn't significant as long as it was in this range). I removed the instances entirely as they wouldn't contribute

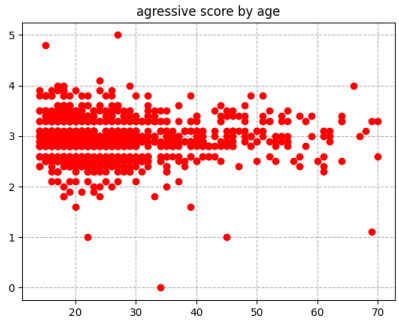

Fig. 6. Agressive score plotted against Age

Fig. 7. Self-defeating score plotted against Age

anything to the training of any model with inferred data, as age is what I want the models to predict.

The gender data was encoded already with 'male' equivalent to 1, 'female' equivalent to 2, and 'other' equivalent to 3. However the data-set contained a small number of gender values set to 0; I decided to group these together with 'other' by setting all the 0s to 3s.

Finally, I had to infer values for instances with missing question answers (set to -1 in the data-set). I used scikit-learn's built in imputer to replace each of these with the mean response for that question. This way, the missing values wouldn't add any bias to the model.

After data cleaning, I began feature construction and transformation. I decided to remove the 4 HSQ score features from the set of inputs as these are directly calculated from the 32 questions, and would therefore otherwise cause multicollinearity (a problem for regression [6]). My input feature vectors therefore consisted of the 32 question responses, the respondent's gender and the accuracy rating they gave their answers. This is heterogeneous data (made up of both numerical and categorical data) meaning I had to transform the categorical data (gender) into numerical data.

Although the gender was already nominally encoded with integers, these integers imply an ordinal nature of gender, which is incorrect. To counteract this I decided to encode the gender feature using one-hot encoding: this meant turning the gender feature into 3 features (one for each gender option) exactly one of which would be set to 1 (the others set to 0) for each instance corresponding to the gender of that instance.

I also had to apply feature scaling to the numerical features (Q1-32, age and the accuracy) to ensure they were all on a similar scale. This ensured that no one feature was dominant in calculating the distance for classifying algorithms, and helps speed up gradient descent for regression algorithms. It also counteracts the skew seen in the age feature. To do this I used standardisation as this worked well with all the algorithms I was using and isn't sensitive to outliers. This works by offsetting the value by the mean of the feature and then scaling it to have unit variance.

As I was using a classification algorithm as well as regression algorithms (explained in the next section), I had to create a version of the output age feature that was categorical in nature rather than numerical. To do this I grouped the ages into 5 quantile categories (each category has the same number of instances - used to counteract the skewed age set) and encoded this using ordinal encoding. I did this with the scikit-learn KBinsDiscretizer encoder (which carries out the process I just described).

Finally I had to split the data into training sets and testing sets so I could effectively evaluate my models after training. I used a test size of 20% of the data-set - this was not so small as to cause overfitting but also not so big as to not effectively train the model (underfitting).

2.3 Algorithm Selection

Due to the (originally) numerical nature of the age feature I decided to choose algorithms that are used to predict numerical data. This meant picking regression algorithms. I decided to train models with 2 different regression algorithms so I could compare my results with each of them. I specifically chose regularisation algorithms which introduce a small amount of bias, to prefer smaller coefficients, which reduces overfitting. I used Ridge Regression to train one model, and LASSO (Least Absolute Shrinkage and Selection Operator) regression to train another. These both perform regularisation in slightly different ways.

I also decided to train a model by grouping the age data into categories. I hypothesised that in the case where the regression algorithms dont perform well, a categorical algorithm may perform better as it is easier for a predicted age value to be in a larger category than it is for it to be close to an exact value. By selecting categories by quantiles, I could also counteract the skew of the age feature. To do this I had to select a classification algorithm - I decided to use a K-nearest-neighbours classifier as I had more than one category to select from.

2.4 Model Training

To train the ridge regression model I used scikit-learn's built in ridge regression algorithm. This carried out gradient descent using a normal mean squared error cost function with added component increasing the cost proportional each of the coefficients squared with proportionality constant alpha. This therefore penalised coefficients at greater distances from the origin (this works as the features have been scaled). If alpha is too large, there is a risk of underfitting as it introduces too much bias. On the other hand if the alpha value is too small there is a risk of overfitting. In order to choose the most effective value of alpha I carried out hyperparameter tuning using leave-one-out-cross-validation for each of a set of predefined candidate alpha values, scoring each alpha value using the mean squared error. I tested 11 different values ranging logarithmically from $10^{-}5$ to 10^5 to maximise the chance of finding an optimal alpha within that range. I carried out a similar process for LASSO regression - I used the built in LASSO algorithm from scikit-learn, which carries out coordinate descent using a mean squared error cost function but instead adds a value directly proportional (constant alpha) to the sum of the absolute values of the coefficients of the model. This tends to favour models with fewer non-zero coefficients and therefore reduces the number of features the age is dependent on. Cross validation is again used to on many values of alpha to find the value which produces the best mean squared error score.

Finally I trained a k-Nearest-Neighbours model using the built in algorithm in scikit-learn. This stores the training data to memory and when predicting new age categories, the model finds the most common category out of the k nearest instances (distance calculated using the sum of the squares of each of the differences in features within the feature vector). I used grid-search-cross-validation to tune the hyperparameter k, to give the optimum accuracy (proportion of correct predictions out of total predictions). I used accuracy as each age category has the same number of instances. The candidate values of k were every 20 integers from 10 to 510 - too high and the model loses its predicting capabilities, too low and the model becomes overfitted to the training data.

2.5 Evaluation

I evaluated the ridge regression model by calculating the built in R^2 score for the fitted model on, the test data - this calculates the mean squared error but each component is scaled down by that equivalent actual value's distance from the mean age. This effectively allows a comparison of the model against just predicting the mean age for each instance. I also calculated the Mean Squared Error for the test data's predictions, as well as the Root Mean Squared Error. Finally I plotted each test data points absolute prediction error against the actual age value. I calculated the same scores for the LASSO regression model as well allowing me to compare the two. Finally I evaluated the kNN classifier using the accuracy score, which is incapable of being compared with the regression scores as kNN is a classifier not a regressor.

3 Model Comparison

The ridge regression model's optimal alpha value ended up as 100 - with this value it achieved an R^2 score of roughly 0.122 meaning it only just beat the mean as an age predictor (a score of 0 would be achieved by the mean). The RMSE score was approximately 9.84 meaning predictions were on average nearly 10 years away from the correct value. The ridge regression model shows a very weak correlation between humour styles and age. Next, the LASSO model's optimal alpha value was approximately 0.0166, and the model achieved an R^2 score of roughly 0.126 - only slightly

better than ridge regression. It had a slightly smaller RMSE of 9.81. The LASSO model resulted in more coefficients equal to zero as predicted. For both of these models I plotted the absolute error values for each test instance against the actual age (Figure 8 and 9):

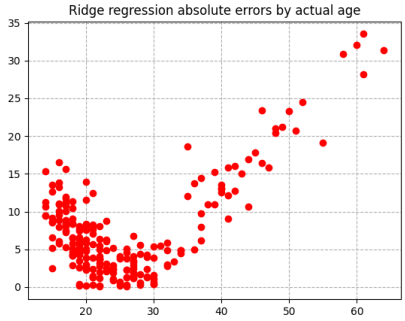

Fig. 8. Predictions are significantly more accurate at lower ages for ridge regression and LASSO regression

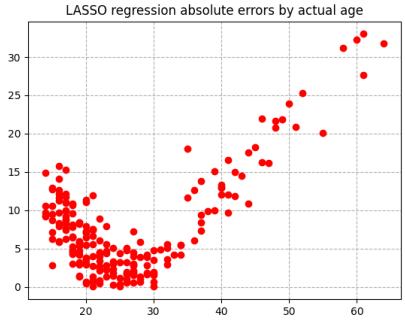

Fig. 9. LASSO

So the regression models do work somewhat better as predictors for lower ages, suggesting that in subsequent training the higher age numbers should be clipped altogether.

Finally for the kNN model, the optimal k value was 90, achieving a score of 0.254. Again this is only slightly better than if the modal category had been used as the age predictor (5 equal sized categories = 0.2 accuracy). So this kNN model is not that good of a predictor for age.

4 Conclusion

In summary, I used the humour styles questionnaire dataset to train 3 different age predicting models, at varying degrees of success. All 3 models showed a weak but not insignificant correlation on average, however the regression models showed a greater predictive ability for lower ages suggesting that there is not enough data from older age groups to predict them properly, but younger ages can be predicted.

References

[1] R. Martin, P. Puhlik-Doris, G. Larsen, J. Gray and K. Weir, "Individual differences in uses of humor and their relation to psychological well-being: Development of the Humor Styles Questionnaire", *Journal of Research in Personality*, vol. 37, no. 1, pp. 48-75, 2003.

[2] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, vol 12, no 85, bll 2825–2830, 2011.

[3] C. R. Harris et al., "Array programming with NumPy", *Nature*, vol 585, no 7825, bll 357–362, Sep 2020.

[4] J. D. Hunter, "Matplotlib: A 2D graphics environment", *Computing in Science & Engineering*, vol 9, no 3, bll 90–95, 2007.

[5] "List of the verified oldest people - Wikipedia", *En.wikipedia.org*, 2022. [Online]. Available: https://en.wikipedia.org/wiki/List_of_the_verified_oldest_people. [Accessed: 24- Jan- 2022]

[6] S. Wu, "Multi-Collinearity in Regression", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/multi-collinearity-in-regression-fe7a2c1467ea. [Accessed: 24- Jan- 2022]