Prediciting happiness from mobile-app survey data

A machine learning approach with comparison of multiple algorithms

Albert Hansa, Alexandros Papageorgiou, Ben Peat
School of Computing
National College of Ireland
Mayor Street, IFSC, Dublin 1

***Abstract*— the goal of this paper is to develop and document a data mining solution to a challenge hosted on Kaggle.com, a platform for predictive modelling and analytics competitions. For this contest, data comes from Show of Hands, a social polling platform developed for use on mobile and web devices. Data is analysed from thousands of users and over one hundred questions in order to compute which responses predict happiness. The data consists of some demographics plus a number of answers to yes/no poll questions. The report also takes a look at what features are the most important for that prediction. The research concludes that is it feasible to achieve a fairly high accuracy in predicting people's happiness status based on their respective answers. Additionally particular ensemble class algorithms associated with bagging and boosting methods tend to perform significantly better in this activity compared to other algorithms tested.**

I. Introduction

A. *Aims and objectives*

This paper discusses a method of objectively predicting the subjective sense of happiness based on informal survey data. While not a scientific study, the aim is to predict how a person responds to "are you happy?" by mining data and, in the process, deducing the most common traits of people who claim they are happy.

The paper analyses data from thousands of users and over a hundred poll questions. Data includes some user demographics; year of birth, gender, income, household status, education level, and party preference among others plus a number of answers to yes/no poll questions from the Show of Hands survey, such as; Are you good at math? Have you cried in the past 60 days? Do you brush your teeth two or more times every day? Mac or PC? Do you drink the unfiltered tap water in your home? Were you an obedient child?

Basically, the objective is to predict the 'happy' variable in the test data using a training model developed from the training set. The R programming language and statistical software was used by the researchers, applying the built in GBM {stats} library and the randomForest {randomForest} package among others.

B. *Challenges and limitations*

Given the data has 4619 observations of 110 variables, one key challenges is to decide which variables really matter in the analysis and prediction of the binary response {happy, unhappy}.

Ability to efficiently handle decision boundaries in the high dimensional feature space is another important consideration. Machine learning algorithms needed to be selected by virtue of their ability to deal with both categorical and numerical data for the purpose of classification.

C. *Evaluation criteria*

The evaluation metric for this competition is AUC, a commonly used evaluation metric for binary problems like this one. The AUC is based on the so-called class probabilities, i.e. the estimated probability of each observation to belong to the class Y = 0 or Y = 1, respectively. The AUC metric is less affected by sample balance than accuracy. A perfect model will score an AUC of 1, while random guessing will score an AUC of around of 0.5. While additional metrics are reported, AUC is considered the standard metric of model comparison for this analysis and the goal is to consistently perform significantly better than the random guess AUC score.

II. Literature review

In the quest for the most accurate predictors of happiness, several research papers were reviewed, evaluated and analysed in order to overcome the anticipated difficulties of dealing with highly subjective data:

- What survey questions to include?
- What variables to keep?
- What methods to choose from?

In the World Happiness Report 2015 six key variables have been used: GDP per capita, social support, healthy life expectations, freedom of life choices, level of corruption, public generosity [1]. Using pooled OLS (ordinary least-squares), regression researches came up with a nearly normal distribution of happiness for the whole world. The model's predictive power is 0.741 (adjusted R-squared). While very valuable and insightful, this type of research is dependent on finding the key variables that define happiness. In the case of this paper, the goal is to predict happiness with maximum accuracy based on very subjective sample data along with some demographic variables.

Several researches have produced statistical models and machine learning techniques with varying degrees of prediction accuracy and interpretability. Louise Millard, opts for PCA (Principal Component Analysis), which is good at finding hidden data patterns in high-dimension data. She also uses lasso (least absolute shrinkage and selection operator) as a regression analysis method that performs both variable selection and regularisation in order to improve prediction accuracy [6].

Millard considers decision trees (except ID3 and C4.5 algorithms as they are classifiers and cannot be applied to regression tasks), support vector machines (SVM are a powerful and flexible method for regression problems) and the k-nearest neighbours algorithm (KNN). It was found that KNN is useful for imputing missing values. For example, using several educational variables to impute the missing values of another education variable. On the other hand, knn is affected by irrelevant attributes and thus prone to errors which can be quite large, so to reduce the error we remove the irrelevant attributes and minimize the number of values that need imputing [6].

A common obstacle when attempting to increase the accuracy rate of predictions is that of missing values. The research by Theresia Ratih Dewi Saputri shows that despite SVM algorithm efficiency and effectiveness, with the improved feature selection (different for every country they observed) the result was still inadequate. A more reliable approach for dealing with missing data is required [13].

The approach discussed in this paper differs to the above mentioned cases since the multiple imputation techniques applied dealt with missing data successfully.

In a study of social intelligence conducted by Takashi Kido, deep learning (neural networks) outperformed all other algorithms (Random Forest, SVM and others. Neural networks produced an accuracy rate of 70.8% [12].

While neural networks are a very popular prediction method, they were not implemented in this paper due to time constraints.

III. Methodology

A. Descriptive statistics & exploratory analysis

The complete dataset contains 4619 observations with a total of 110 variables.

- All but three of the variables are of a categorical nature and there is presence of both nominal and ordinal variables.
- The vast majority of the variables were three-level factors in the form of questions were the answer is "Yes", "No" or "No answer".
- The dependent variable is "Happy", which is of a binary class. The majority class is positive to happy with a small margin i.e. 56 % versus 46%.

Strong correlations between any of the numerical variables and the outcome variable were not identified. Figure 1, as an example illustrates the relationship between age (both as numerical and as factor) and happiness. Some degree of association between seniority and happiness can be identified.

In the dataset there was a characteristic lack of numerical variables to compute paired correlations for. As an alternative technique a variable importance index was used, as a side output of the Random forests algorithm, applied on the original dataset.

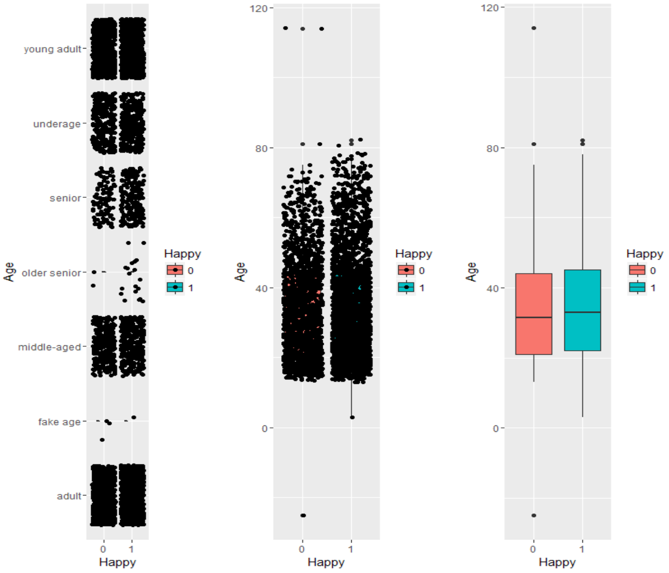


Figure 1 Age/age-group association with happiness outcome

The results imply a strong association of the outcome variable with several predictors, the following in particular: (i) income, (ii) educational level and (iii) the answer to question Q118237: “Do you feel like you are "in over-your-head" in any aspect of your life right now?”

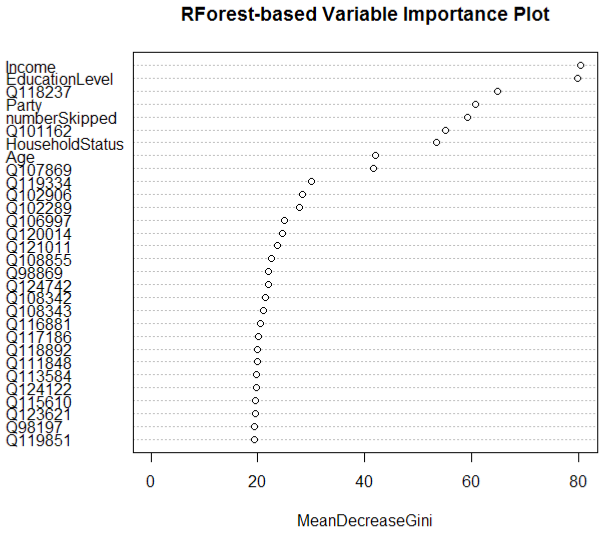


Figure 2 Variables ranked by importance in predicting outcome

B. *Methodology and Algorithms*

A variety of machine learning algorithms were deployed to address the kaggle competition prediction challenge.

Those algorithms were selected based on their capability for dealing with both categorical and numerical data in high dimensionality for the specific purpose of binary classification. The selection deliberately included methods that involve different algorithmic approaches, with various degrees of complexity and interpretability. The main algorithms used to train the data were the following: Naive Bayes, Decision Trees (CART), Random Forests, Gradient Boosting and Support Vector Machines.

Naive Bayes is a typical method used when the data includes many categorical variables, so it was considered as the first standard choice given the type of data involved.

Decision Trees (CART), Random Forest, Gradient Boosting are all methods based on the generation of tree models. Decision trees involve essentially the growing and pruning of a single tree, the other two methods are based on the generation of ensembles of trees. This means that they create synergies by combining the outcomes of multiple base models. The latter are averaged based on a vote of majority to provide reduced variability. For this reason they tend to perform better than the single decision tree method [9]. GBM is a method that fits complicated models by re-fitting sub-models, typically decision trees, to residuals and it is an example of a boosting inspired classifier [10]. Random Forests on the other hand are bagging inspired. One disadvantage of using these two methods is that they are not as intuitive to explain, visualize and interpret.

Support vector machines are known for their capacity to deal with both numerical and categorical data. They are often referred to as “black box” type models and the reason for their selection was to have a reference as to the predictive performance of a complex method, to use as a comparison benchmark for the other algorithms.

IV. IMPLEMENTATION

The implementation involved two main parts: preprocessing including feature engineering and then the subsequent application of multiple algorithms. The objective was to analyse their performance across various metrics and then compare their relative strengths and weaknesses.

A. *Preprocessing and feature engineering*

Given the high dimensionality and the particularities of the dataset i.e. high number of categorical variables and a small number of numerical ones, a series of steps were followed in order to prepare the data accordingly and facilitate efficient algorithm implementation. To ensure reproducibility of this research, the main steps are described below.

1) *Year of birth variable*

This variable is recognised as playing a key role in the state of happiness of individuals. In order to transform the year of birth into a more informative attribute it was converted first to age and then as age it was divided into 7 range-based groups, in line with commonly used standards for labelling age groups (e.g. young adults, middle-aged, senior, etc.). Moreover all irrational ages (for example 0 or 120) were labelled as untrue.

2) *NA values*

There were a total of 684 NA values in the dataset. Many learning algorithms cannot be implemented while NA values are present. Even though it is possible to eliminate records with missing data, it would lead to a considerable amount of information loss. Multiple imputation was used to replace missing data with reasonable values based on repeated Monte Carlo simulations [11].

3) *Skipped questions*

Some respondents of the survey simply skipped certain questions; in this case the response was registered as blank. The fact that a respondent deliberately chooses not to provide an answer to a particular question has some informative value in its own right. Those cases were turned into an additional categorical level labelled as "skipped".

4) *Feature engineering*

A variable was added in the dataset that reflected the total number of skipped questions for every respondent, as this could provide a useful additional signal. Additionally, based on the variable importance score of all the variables in the dataset the top ten questions with the highest score were selected. A new variable was introduced to reflect the total number of "positive-spirit" answers that respondents provided.

5) *About feature selection.*

Feature selection was initially attempted by generating a variable importance table that included the key predictors. However it was decided not to use the tableu, given the context of the data and the question. The pre-processed dataset only included two numerical variables which did not show any strong or even moderate correlation. Moreover, most of the algorithms deployed, such as random forests and gradient boosting were capable of performing automatic feature selection during the training stage [12].

B. *Predictive modelling process*

After pre-processing and feature engineering routines were completed, a final choice of algorithms was decided and the ensuing process was followed with respect to algorithm implementation.

- The data set was split into train and test set on a 70-30 % ratio. A sampling method was deployed to preserve the distribution of the outcome variable in both the train and test sets for improved chances of model generalisability.
- Where needed, special variable transformations were performed. For example, factor variables into dummy variables or conversion of the response variable into binary or categorical outcome as per the individual algorithm specifications.
- As an additional step in the case of the support vector machines, the numerical variables were centred and scaled in line with the respective specifications.
- Next the selection of the various tuning parameters took place by partially exploring the search space. This was achieved via a model specific grid with promising combinations of the parameters, including boosting iterations and shrinkage for GBM, cost for SVM and complexity parameter for CART.
- After evaluating the properties of the model, final predictions were performed on the test dataset and estimated metrics of accuracy in the out of sample data were generated.

The main interface used to access the algorithm properties was the caret package of the R language for statistical programming. This provided a unified framework to access the algorithms, with streamlined model tuning using resample methods and the ability to use parallel processing [12].

V. Evaluation

A. *Evaluation Approach*

The most important success metric is how effective and accurate the model is at predicting unseen data. Testing the algorithm on train data will typically provide over-optimistic accuracy estimates. The commonly used method to overcome the risk of overfitting is cross validation. The particular variation of the method deployed was a repeated 10-fold cross validation with three repeats, which essentially produced a bag of 30 resamples per model [13]. Then, in order to have a good sense of how generalisable the model is, its performance was evaluated on test data.

B. *Evaluation Metrics*

There is a variety of different metrics available for evaluating the quality of predictions. Accuracy, one of the most widely used ones, represents the number of instances that are correctly classified. However, depending on the nature of the problem, it might not be the metric with the highest practical significance. The current analysis involves prediction on a binary class, "happy", "unhappy". Under this scenario typical measures of evaluation include metrics like sensitivity, specificity, and Area Under the Curve (AUC). These metrics can be used individually or combined depending on the nature and context of the data and the question.

The “what predicts happiness?” Kaggle competition indicated that the formal evaluation metric to be used is AUC [14]. The AUC score can be thought of as the probability that given a positive and negative instance, the classifier can correctly classify both. While additional metrics are reported, AUC is considered the standard metric of model comparison for this analysis.

Overall, there are two dimensions considered when evaluating and comparing models. The first is the degree to which each model independently betters the “Null” (Base-line) model, which corresponds to a random guess. Additionally, the numerous models were compared among themselves in order to identify if one of them consistently outperformed the rest and therefore could be considered as the best fit for the given problem.

C. *Results*

The models were evaluated in two stages. The first one involved comparison of the train dataset, based on multiple resamples. Figure 1 suggests that the complex models perform better compared to a decision tree. Those models perform similarly in terms of AUC scores. Gradient boosting exhibited the highest average score and least amount of variation.

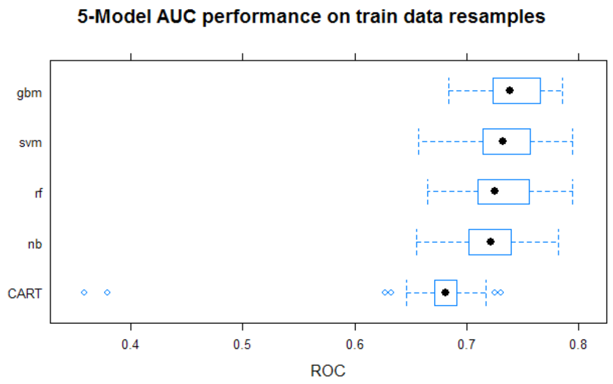


Figure 3 Performance Comparison on the train data

After the models were tuned, five respective model fits were used to make predictions on the test data. Classes and probabilities were computed and ROC curves were generated along with the respective AUC scores.

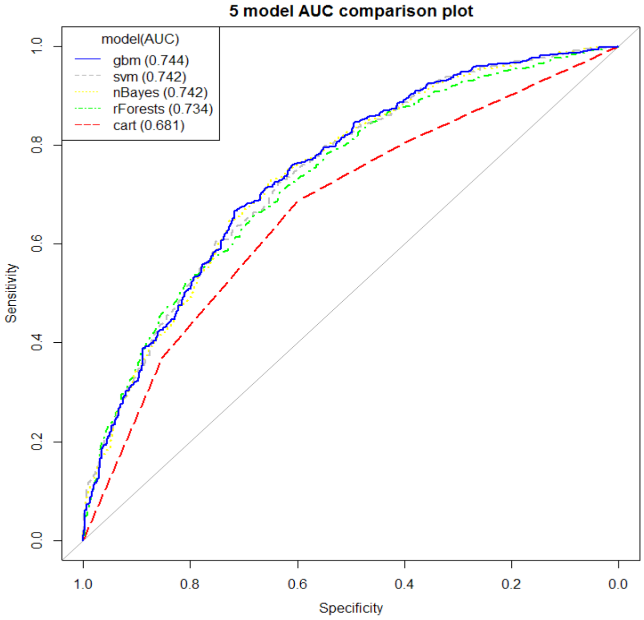
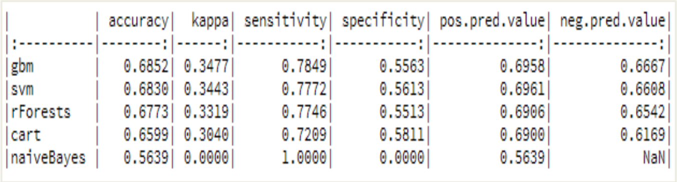


Figure 4 AUC Performance comparison on test data

Figure 2 illustrates graphically the results. Not unlike the train set performance, Gradient Boosting (GBM), Support Vector Machines (SVM), Random Forests (RF) and Naïve Bayes (NB) score highly, with decision tree performance notably lower. GBM had the highest AUC score, by a small margin. Given the pre-determined performance criterion, the GBM model could be considered the most effective in terms of addressing the Kaggle challenge.

Additional metrics of performance, including accuracy, sensitivity and specificity are presented in table 1. Accuracy-wise, GBM is the leading algorithm again with a score of over 68%. Naïve Bayes algorithm on the other hand returned a one class prediction, which resulted in the lowest accuracy corresponding to the null model.



	accuracy	kappa	sensitivity	specificity	pos.pred.value	neg.pred.value
gbm	0.6852	0.3477	0.7849	0.5563	0.6958	0.6667
svm	0.6830	0.3443	0.7772	0.5613	0.6961	0.6608
rForests	0.6773	0.3319	0.7746	0.5513	0.6906	0.6542
cart	0.6599	0.3040	0.7209	0.5811	0.6900	0.6169
naiveBayes	0.5639	0.0000	1.0000	0.0000	0.5639	NaN

Table 1 Multiple metrics comparison of the 5 algorithms used

D. Discussion of results, interpretation and implications

The results described above suggest that the deployment of learning algorithms for the question at hand provide a fairly effective solution to a problem pertaining to human behaviour, which is inherently difficult to predict.

All algorithms consistently performed better than a random guess, or even a uniform response based on the majority class proportion. The algorithm that performed best across all metrics was GBM. It is important to highlight however that accuracy and AUC scores are not always the only consideration. This might be the case when other factors can play a role too like need for simplicity of prediction or use of fewer variables [15]. Meeting those conditions typically make a model easier to maintain over new data, more practical to interpret, scalable etc. In this case a decision tree model, such as the one presented for illustrative purposes in Figure 3 might be worth considering

In terms of a more practical interpretation, the use of the variable importance feature which is integrated into the random forests algorithm suggested that answers to questions regarding whether a person is optimist or whether he or she feels normal can be a strong signal with respect to the final happiness outcome, alongside other historically recognised factors such as educational level and income. This fact, in conjunction with the pluralistic approach to algorithm implementation and comparison can be perceived as the main contribution of this research towards a solution to the happiness prediction problem.

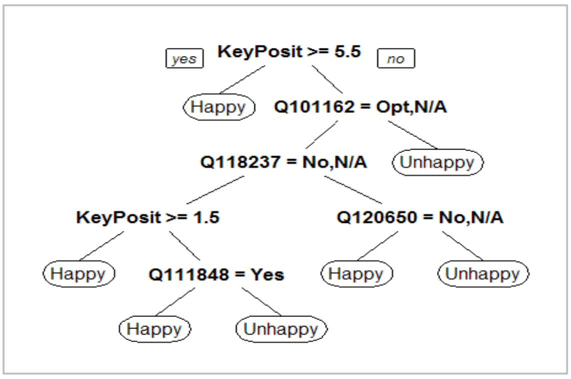


Figure 5 Visualisation of the tree model based on train data

VI. Conclusion

Over the course of developing a data mining solution for predicting happiness based on informal survey data, the researchers start out with some simple models and finish by concluding that complex algorithms associated with bagging and boosting techniques tend to perform significantly better.

A deeper understanding of the problem, leveraged through previous research, guided preprocessing and feature engineering tasks. Moreover, because of the nature of the survey, creative solutions were brought to bear in order to help account for missing values, wrong-fake values, and skipped survey questions. From a technical point of view the researchers performed extensive feature engineering and employed multiple algorithms (both simple and complex) that performed very well on the out of sample data.

Despite the subjectivity of 'happiness' measurements, using a sound approach and the right data mining techniques, "elusive" happiness is perhaps less abstract and intangible as has been previously thought to be the case. The results discussed in this paper are in line with the 'Easterlin paradox', suggesting that the link between a society's economic development and its average level of happiness is less obvious than perhaps previously thought to be. The researchers deduce that survey questions with high subjectivity can nonetheless be meaningful in data mining research.

The inevitable limitations of this work stem from the nature of the original dataset, data comes from *Show of Hands*, a social polling platform developed for use on mobile and web devices and was collected on a voluntary basis. The dataset cannot be considered as representative of the total population.

For future work, the researchers believe that improving accuracy would necessitate further parameter tuning of the 5 algorithms; implementation of algorithms such as neural networks and xGBoost; and, working on ensemble techniques to combine the results of more than one algorithm together for improved prediction accuracy.

VII. References

[1]John Helliwell, Richard Layard, Jeffrey Sachs, *World Happiness Report* 2015, pp14-42,

[2]Rafael Di Tella, Robert MacCulloch, Some uses of happiness data in Economics, *Journal of Economic perspectives*, 2006

[3]Robert MacCulloch, *Can* "happiness data" evaluate economic policies?, Auckland University, 2016

[4]David G. Blanchflower, Andrew J. Oswald, Money, sex and happiness: an empirical study, *Scandinavian Journal of Economics*, 2004

[5]Louise Millard, *Data Mining and Analysis of Global Happiness: A Machine Learning Approach*, University of Bristol, September 2010, pp 41-53

[6]Nwamu Phillippa Unoma, *Learning the student's happiness model*, The University of Manchester, October 2013

[7]Takashi Kido, Melanie Swan, *Machine learning and personal genome informatics contribute to happiness sciences and wellbeing computing*, Riken Genesis Co., Tokyo, Japan, 2016, p 5

[8]Theresia Ratih Dewi Saputri, Seok-Won Lee, "Are we living in a happy country: an analysis of national happiness from machine learning perspective", Ajou University, South Korea, 2009, p 3

[9] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[10] S. B. Taieb and R. J. Hyndman, "A gradient boosting approach to the Kaggle load forecasting competition," *Int. J. Forecast.*, vol. 30, no. 2, pp. 382–394, 2014.

[11] R. Kabacoff, *R in Action: Data Analysis and Graphics with R*, 2 edition. Shelter Island: Manning Publications, 2015.

[12] M. Kuhn, "Building Predictive Models in R Using the caret Package," *J. Stat. Softw.*, vol. 28, no. 5, Nov. 2008.

[13] "The 5th Tribe, Support Vector Machines and caret," *Revolutions*. [Online]. Available: http://blog.revolutionanalytics.com/2015/10/the-5th-tribe-support-vector-machines-and-caret.html. [Accessed: 17-Jul-2016].

[14] "Evaluation - The Analytics Edge (15.071x) | Kaggle." [Online]. Available: https://www.kaggle.com/c/the-analytics-edge-mit-15-071x/details/evaluation. [Accessed: 18-Jul-2016].

[15] N. Zumel and J. Mount, *Practical Data Science with R*, 1 edition. Shelter Island, NY: Manning Publications, 2014.