Principles & Elementary Models

An Investigation into the Socio-Economic Factors that Drive Happiness and Unhappiness

AZEVEDO COSTA Gabriel

TABLE OF CONTENT

Introduction

Given the myriad of factors that can influence an individual's happiness, we will look to discover which socio-economic aspects play a pivotal role. Can we predict an individual's happiness level based on these factors? If so, how accurately?

With the growing interest and people prioritizing well-being, determining what leads to happiness has become an increasingly important research matter. For policymakers, it can guide the formulation of policies that enhance well-being. For individuals, it offers a reflection on the aspects of life that might be worth prioritizing. Moreover, businesses can leverage these insights to foster a happier workforce, leading to increased productivity and reduced turnover.

We will be examining the interdependence between different economic and social components on people's subjective well-being. From a comprehensive dataset of thirty-three variables such as demography, status and behavior, we will construct a predictive model of happiness that will help us understand the relationship between the gathered data and happiness.

Data Collection, Preparation and Methodology

1) Data Generic Information

The dataset chosen was sourced from the Comprehensive R Archive Network repository. It originates from a survey that aimed to gauge the happiness levels of individuals in America based on various socio-economic factors. Each row in the dataset represents a person and provides comprehensive details about their social situation, habits and various key economic factors.

2) Data Preprocessing

2.1) Data Cleaning

The initial dataset contained 33 columns but several were either redundant or not directly relevant to the study's objective. These columns, such as encoded year and region, were dropped to streamline the dataset and focus only on the current socio-economic factors.

2.2) Handling Missing Values

NaN values in various columns were addressed based on the context. For instance, NaN values in the 'income' column were interpreted as no income and were filled accordingly with 0. The same

logic led us to fill the other columns with 0 such as “educ” meaning the person had no specific education, “babies”, or “tvhours”.
For the “workstat” column we considered that NaN values referred to unemployed people so we filled in the missing values by mentioning them.

2.3) Encoding

The categorical variables such as “income”, “workstat” or “attend”, were manually encoded to numerical values by creating a dictionary with the numerical values to assign depending on the modality. This encoding was done based on the inherent order or significance of the categories. The remaining columns were already encoded or were binary.

2.4) Feature Engineering

New features were derived from existing ones to capture more nuanced information. For example, a 'gwbush' column was created by merging information from two columns related to voting for G.W. Bush in different years. We decided to not consider voting in different elections so made it a single binary column representing whether an individual had voted in for Bush either election (inclusive or).

Also, we noticed that the “vhappy” column representing whether the person is very happy or not was created using the happy column. So, to have a different interesting target we created a nothappy column using the same process.

The purpose of all these transformations is to give a more comprehensive and exploitable target for analysis, by sorting if the person is very happy, not happy or moderately with binary columns for example. This will help identify any potential patterns or trends related to what can influence and have an impact on happiness.

2.5) Principal Components Analysis

Once our features and dataset were ready for further analysis, we conducted a PCA analysis to try to reduce dimension and attempt to find any underlying patterns. As we still had over 20 dimensions we thought it would be interesting to see to what extent we could reduce this dimensionality compared to the amount of explained variance and therefore information we lose.

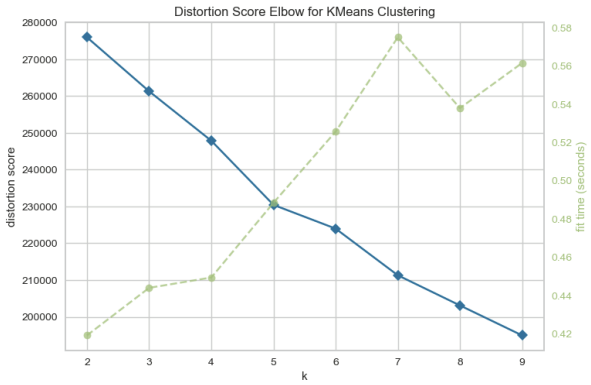
First, we prepared the data by standardizing the non-binary columns and then trying to select an appropriate amount of factors to represent the whole dataset.

After that, we run several classifier models with both targets “vhappy” meaning the person is very happy and “nhappy” meaning that the person is not happy separately. The aim was to discover whether we could accurately predict these outcomes. The specific models used are Logistic Regression, DecisionTree Classifier, KneighborsClassifier, Random Forest Classifier and SVC. We then tuned the hyperparameters to see whether we could

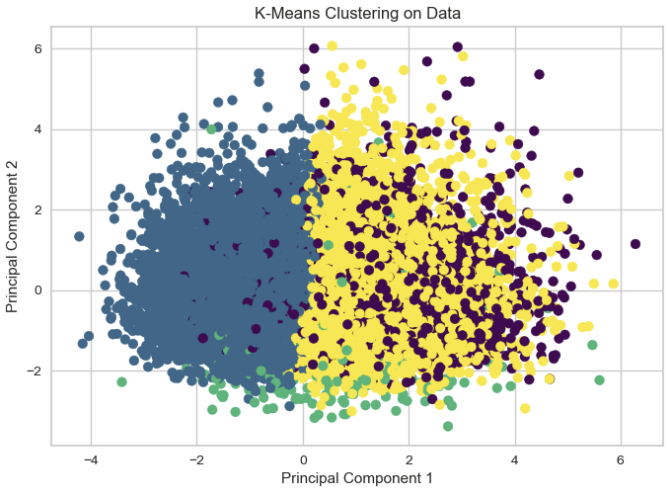
2.6) Clustering

Clustering is used to find structure in a set of unlabeled data by organizing objects into groups whose members are similar in some way. A cluster is therefore a collection of objects that are "similar" to each other and "different" from objects belonging to other clusters. We tried to apply this method to our data using 3 different clustering processes: Kmeans, DBSCAN and agglomerative clustering. It's important to perform a PCA before starting clustering to reduce the dimensionality of the data. For Kmeans clustering, it's important to define the number of clusters. For this purpose, we used the elbow method. Once an appropriate number of clusters has been found, we can use the function to create a graphical representation of the clusters. We can see that four clusters have been created, although their boundaries are not well-defined.

Regarding DBSCAN, we need to define the epsilon before we can use the function properly. To do this, we used a k-distance plot. Then, as with Kmeans, we create the graph with the clusters. For this method, we can see that the results are not good: clusters overlap, boundaries don't make sense, and some clusters have very few points. Finally, we also used the agglomerative cluster method. To use the function, you need to know the number of clusters. Here we've just used the result of the elbow method used in Kmeans.



Before jumping into the clustering we kept in mind that it might not be proper to use it in our specific case, given the data set that we have and information that we want. Here we can see on that graph above that the Elbow method is struggling to define the optimal number of clusters, this might be due to the large amount of binary features that we have. We choose to use the number of cluster = 4 in order to continue the process.



The graph we have before is the visual representation of our clusters using Kmeans. We can see that it is really messy, our clusters are not properly defined. In fact they seemed to be all merged in one space. That is not a great result and doesn't help to figure a pattern within the data set. Even worse results were also observed when using DBSCAN and Agglomerative Clustering whereby the clusters do not provide any meaningful results.

Machine Learning

The aim here is to build a model which can effectively predict the outcome of unseen data. We will go through several classifier models as our outcome is binary and will compare their efficiency using the following key performance indicators:

- **Accuracy:** refers to the proportion of correctly classified data points with the total number of instances. It is a simple metric that measures the overall performance of a model in terms of correctly predicted outcomes.
- **ROC AUC**: evaluates the model's ability to discriminate between positive and negative classes. It quantifies the model's ability to rank observations and is particularly useful when the dataset is imbalanced.
- **F1 Score:** combines precision and recall to provide a single measure of a model's performance in binary classification tasks. It considers both false positives and false negatives and is useful when the class distribution is uneven.
- **Kappa Score:** measures the agreement between observed and expected classifications, taking into account the possibility of the agreement occurring by chance. It is commonly used to assess the performance of a classification model beyond what would be expected by random chance.

3.1) Model Training

To train our model efficiently with many classifiers and with all the preprocessing steps we created a comprehensive class that automates all the steps and prints the results.

The class that we named Classifier, is composed accordingly:

The split_and_stratify method splits the input data into training and testing sets called X_train, X_test, y_train, and y_test. We also used the stratify parameter to ensure that the distribution of values in the dataset is preserved in the train and test datasets. When we have imbalanced classes in the dataset, using the stratify parameter helps in creating training and testing sets that have the same proportion of classes as the original dataset.
In the case of our binary classification problem where the target variable has two classes, '0' and '1', the stratify parameter ensures that both the training and testing datasets have approximately the same proportion of '0' and '1' as the original dataset. This is important because it helps prevent the model from being biased towards the majority class and ensures that it learns to predict both classes effectively.

Then the training method fits the provided model to the training data and uses the fitted model to make predictions on the test data. The predicted values are stored in the class attribute that we named y_pred.

Finally, the scoring method computes various evaluation metrics, including accuracy, F1 score, and, if the binary parameter is set to True, it also calculates Cohen's Kappa and ROC AUC scores.

3.2) Interpretation of the results for “vhappy”

We ran multiple classifier models on the same dataset, so we can compare their performances and identify the model that best suits our goal. That way we can also assess the robustness and reliability of the predictions across different approaches. This helped us make more informed decisions and reduce the risk of relying on a single model that might be sensitive to certain types of data or noise.

As mentioned before we chose to run the following models: Logistic Regression, DecisionTree Classifier, KneighborsClassifier, Random Forest Classifier and SVC.

Here are the results obtained that will be interpreted separately:

Model	Accuracy	ROC AUC	F1	Kappa	Time taken
LogisticRegression	0.700	0.675	0.143	0.073	0.550
DecisionTreeClassifier	0.603	1.000	0.364	0.076	0.082
KNeighborsClassifier	0.645	0.785	0.294	0.072	0.383
RandomForestClassifier	0.692	1.000	0.265	0.120	1.837
SVC	0.693	0.569	0.000	0.000	33.857

3.2.1) Logistic regression

We first performed a Logistic Regression on our data. With an accuracy of 0.700, it means that the model correctly predicts whether a person is very happy or not for approximately 70% of the data points in the dataset. The ROC AUC of 0.675 means that the model's ability to distinguish between those who are very happy and those who are not is moderate, but it may not be very effective at ranking individuals based on their happiness levels. The F1 score of 0.143 indicates that the model struggles to strike a good balance between precision and recall, effectively capturing both true positives and true negatives. However, the Kappa score of 0.073 suggests that there is only slight agreement between the predicted and actual happiness levels, implying that the model's performance is only slightly better than random chance.

3.2.2) Decision Tree Classifier

The Decision Tree Classifier resulted in an accuracy of 0.603, indicating that it correctly categorized around 60% of the individuals based on their happiness levels. This suggests that the model's predictive performance is not as reliable as desired. The perfect ROC AUC score of 1.000 suggests that the model is capable of precisely distinguishing between those who are very happy and those who are not, but it may have overfit the training data or incurred data leakage. The F1 score of 0.364 suggests that the model struggles to strike a balance between precision and recall, effectively managing both false positives and false negatives. The Kappa score of 0.076 indicates a fair agreement between the predicted and actual happiness levels, suggesting that the model's performance is better than random chance but could still benefit from further enhancements.

3.2.3) KNeighborsClassifier

The KNN Classifier got an accuracy of 0.645, indicating that it correctly categorized approximately 64% of the individuals based on their happiness levels. This suggests that the model's predictive performance is moderately reliable but has also room for improvement. The ROC AUC of 0.785

suggests that the model has a good ability to distinguish between very happy individuals and those who are not, implying it is relatively effective at separating the two categories. The F1 score of 0.294 suggests an imbalanced performance between precision and recall, indicating that the model cannot make accurate predictions while maintaining a good balance between false positives and false negatives. The low Kappa score of 0.072 suggests only slight agreement between the predicted and actual happiness levels, indicating that the model's performance is not significantly better than random chance.

3.2.4) Random Forest Classifier

The Random Forest Classifier got an accuracy of 0.692, suggesting that it correctly categorized approximately 69% of the individuals based on their happiness levels. This indicates that the model's predictive performance is reasonably reliable and better than some of the other models analyzed. The perfect ROC AUC score of 1.000 suggests that the model has excellent discrimination ability, meaning it can effectively distinguish between very happy individuals and those who are not without errors but this could be attributed to data leakage. The F1 score of 0.265 implies that the model struggles to strike a balance between precision and recall, failing to ensure a good trade-off between correctly identifying true positives and avoiding false negatives. The Kappa score of 0.120 suggests a fair agreement between the predicted and actual happiness levels, indicating that the model's performance is better than some other models but may still benefit from further improvements.

3.2.5) Support Vector Classifier

Finally, the SVC achieved an accuracy of 0.693, suggesting that it correctly categorized approximately 69% of the individuals based on their happiness levels. This indicates that the model's predictive performance is relatively reliable but may not be the most accurate compared to other models. The low ROC AUC score of 0.569 suggests that the model's ability to distinguish between very happy individuals and those who are not is limited, indicating that the model may struggle with effectively ranking individuals based on their happiness levels. The F1 score of 0.00 suggests that the model fails to maintain a balance between precision and recall, suggesting that it cannot make accurate predictions without compromising on either false positives or false negatives. However, the Kappa score of 0.000 indicates no agreement between the predicted and actual happiness levels, suggesting that the model's performance is not better than random chance.

3.2.6. Conclusion

Based on the analysis of the performance indicators for each model, we can make a conclusion regarding which model to retain for predicting happiness levels in the dataset.

Considering the key metrics and their implications, the Decision Tree Classifier and Random Forest Classifier stand out as the most promising model. It showed a strong balance between accuracy, discrimination ability, and overall reliability compared to the other models. However, further tuning and optimization of the Random Forest model could potentially enhance its performance and predictive capabilities.

3.3) Interpretation of the results for “nhappy”

As mentioned before, we created a new binary variable representing if a specific person is not happy as our original variable representing happiness had 3 different modalities: Not happy, moderately happy and very happy. That way we have a new variable that is worth analyzing as a target to get more valuable insights on our gathered data.

We ran the same models as for the other outcome and here are the results obtained that will be interpreted separately again:

Model	Accuracy	ROC AUC	F1	Kappa	Time taken
LogisticRegression	0.879	0.776	0.023	0.018	0.615
DecisionTreeClassifier	0.791	1.000	0.246	0.127	0.067
KNeighborsClassifier	0.863	0.875	0.056	0.018	0.313
RandomForestClassifier	0.877	1.000	0.054	0.039	1.330
SVC	0.878	0.607	0.000	0.000	26.498

3.3.1) Logistic regression

The Logistic Regression model got a great accuracy of 0.879, indicating that it accurately classified approximately 88% of the instances as either not happy or otherwise. The ROC AUC of 0.776 suggests a good ability to distinguish between the positive and negative classes, and with some

room for improvement. The F1 score of 0.023 indicates an imbalanced performance between precision and recall, preventing reliable predictions. The low Kappa score of 0.018 implies only slight agreement between predicted and actual values, suggesting that the model's performance is marginally better than random chance.

3.3.2) Decision Tree Classifier

The Decision Tree Classifier had an accuracy of 0.791 and correctly classified close to 79% of the instances as not happy or otherwise. The ROC AUC score of 1 means perfect discrimination ability, so the model does not make mistakes in separating positive and negative classes, this could have occurred due to data leakage. A marginally balanced performance between the precision and the recall as reflected by the f1 score of 0.246. The Kappa score (0.127) represents a fair agreement between predicted and observed values, showing that the model performs better than random chance.

3.3.3) KNeighborsClassifier

KNN Classifier had an accuracy of 0.863 meaning it identified about 86% of the instances as not happy. ROC AUC of 0.875 shows good competence in separating positive and negative classes. The F1 score of 0.056 indicates a poorly balanced performance between precision and recall, implying inaccurate predictions and an imbalance between false positives and false negatives. The Kappa score of 0.018 indicates little agreement between predicted and actual values. This means that the model does not perform significantly better than random chance.

3.3.4) Random Forest Classifier

The Random Forest Classifier gave an accuracy of 0.877, which meant 88% of the instances were not happy or otherwise. The highest ROC AUC value of 1.000 means that the model can perfectly discriminate between positive and negative classes with no errors. However, this could have occurred due to data leakage. This F1 score of 0.054 indicates a poor trade-off between precision and recall, implying that the classifier was unable to accurately detect true positives and prevent false negatives. The Kappa score of 0.039 implies that the predicted and actual values are in fair agreement, and the model performs slightly better than random chance.

3.3.5) Support Vector Classifier

SVC had a high accuracy of 0.878, meaning it correctly categorized around 88% of the instances as not happy or otherwise. The ROC AUC of 0.607 means that the model’s ability to differentiate between the positive and negative classes is moderate, with scope for improvement in effectively separating the two groups. The F1 score of 0.00 indicates a poor balance between precision and recall while the Kappa score of 0.000 shows that the actual and predicted values did not agree, thus, the model’s performance is not better than random chance.

3.3.6) Conclusions

Based on the analysis of the performance indicators for the “nhappy” target variable for each model, the Random Forest Classifier and Decision Tree Classifier proves to be two of the best ones for predicting whether one is not happy. It had the best trade-off between accuracy and reliability. However, this is not without its caveats such as the potential data leakage or overfitting that cause the models’ ROC AUC to be 1. Despite this, we can still make additional adjustments that might improve the model’s performance and forecasting capabilities.

Interestingly, we noticed that we had better performances when choosing the variable ”nhappy” as a target instead of “vhappy”. This can be explained because of the sizes of the different variables that were much higher for vhappy as a target variable compared to nhappy. This suggests that the variables used were much better in determining someone’s unhappiness compared to their happiness.

3.4) Model Tuning

After applying GridSearchCV for model tuning on the Random Forest Classifier for predicting vhappy, the best hyperparameter configuration was determined to be:

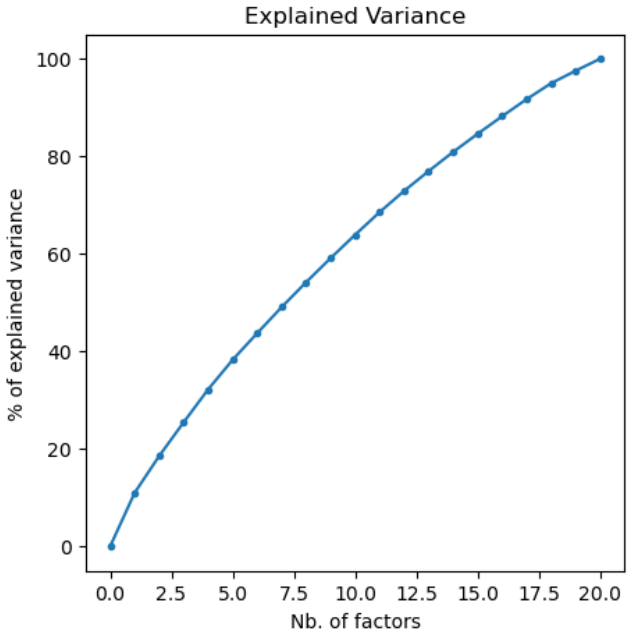
- Bootstrap: False
- max_depth: 10
- max_features: 7
- min_samples_leaf: 5
- min_samples_split: 2
- n_estimators: 30

Setting bootstrap to “False” means that the trees were built upon the whole dataset. Setting “max_depth” as 10 means that every tree in the forest has a maximum depth of only 10 and so prevents the overfitting that comes with too many nodes. The choice of 7 max_features indicates that only 7 of the potential features to be considered in looking for the best split will be taken into account to simplify each of the trees. A more robust and generalizable model is obtained through min_samples_leaf and min_samples_split of 5 and 2, respectively, which represent the minimum number of samples required to be at a leaf node and the minimum number of samples required to split an internal node.

This tuned Random Forest Classifier configuration is expected to offer a compromise between reducing overfitting, maintaining model complexity, and increasing the accuracy of prediction. The combination of these parameters implies that the model would be more robust and less biased, thereby providing a reliable prediction of whether a person is very happy or not.

3.5) PCA and Permutation Importance

The PCA results were not especially helpful. 6 seemed a logical amount of factors. The issue was that even then under half of the variance was explained, and as you increase the number of factors variance increases near linearly. See the figure below:



We see this is terrible for PCA - every original factor contributes a roughly equal share of the variance. This could be because much of the data was categorical. We treated it as quantitative as we could transform it to be ordinal with many modalities but this could still impact performance. Alternatively, it could simply be that all the variables contribute important information.

To see if we could measure the impact of each variable we investigated permutation importance using the best model from the modeling section applied with both very happy and not happy as targets. See below for the results of happy and unhappy on the left and right:

var	var_perm
attend_enc	0.01042
income_enc	0.00777
year	0.00390
prestige	0.00371
unem10	0.00251
educ	0.00199
workstat_enc	0.00180
teens	0.00164
tvhours	0.00157
babies	0.00131
divorce	-0.00078
gwbush	0.00067
black	0.00055
widowed	-0.00047
mothfath16	0.00037
owngun	0.00037
preteen	0.00014
female	-0.00012

var	var_perm
tvhours	-0.00094
female	-0.00082
mothfath16	0.00078
educ	-0.00068
gwbush	-0.00057
workstat_enc	-0.00055
year	-0.00054
prestige	-0.00053
owngun	-0.00050
income_enc	-0.00049
unem10	-0.00036
attend_enc	0.00023
preteen	-0.00016
black	-0.00014
babies	0.00014
teens	-0.00013
divorce	0.00002
widowed	-0.00001

This is more insightful - interestingly we see regularity with which people attend religious events, which income category they fall into, and job prestige have a significantly greater impact than the others on those who are very happy. Further, comparatively between very and not happy we see generally a much smaller impact on the latter, possibly suggesting that our variables have a greater impact on happiness than unhappiness.

It must be noted that clearly, the permutation importance calculations will attribute less weight to binary columns with sparse data. If there is minimal information in a column, shuffling it will make less of a difference. As we see the binary columns have all sunk to the bottom for both happiness and unhappiness, but this likely a flaw of this method for this data structure rather than indicative of the impact they have.

Conclusion & Discussions

In conclusion, the comprehensive analysis of socio-economic factors on individuals' levels of happiness enabled us to obtain significant information about our dataset. The predictive models developed from the dataset showed that certain variables, such as income, professional prestige and education, do indeed influence determining happiness. However, the religious attendance variable emerged as a more important determinant than expected, highlighting the multi-directional nature of happiness.

The disparity in the effectiveness of the variables in predicting ‘very happy’ (vhappy) outcomes compared to ‘not happy’ (nhappy) outcomes is particularly noticeable. The variables were significantly more predictive of unhappiness, with the target variable ‘nhappy’ being multiple times smaller than ‘vhappy’. This suggests that the factors leading to unhappiness may be more concrete or consistent across individuals, whereas happiness may be influenced by a wider and more complex set of factors.

A possible error in our data preparation was how we encoded the targets for supervised learning. The nhappy/happy balances (both to each other and against the null values for each) were not good, so a possible better approach would be to try multicategory classification with the ‘kind of happy’ middle target. This could offer a possible explanation as to why various forms of analysis had greater impact when applied to happiness - there were approximately 2.5 times as many

The unexpected results for the 'widowed' status variable similarly reflect a possible issue with our processing of the data. It was the lowest predictor of unhappiness. This observation highlights that the treatment of binary variables and the representation of minority statuses in datasets should be given particular attention in future analyses to avoid potential bias and misinterpretation.

In addition, the low and sometimes zero values for kappa for all models could show that the models are not better than random chance in terms of determining one’s happiness or unhappiness. This could be attributed to the fact that many other variables affect happiness which were not or could not be surveyed during the data collection process. This thus shows a limitation of attempting to determine happiness using survey data.

Looking forwards, these results open up many areas for discussion and research. Why could this be? Intuitively factors which characterize organized religion such as community, purpose and a sense of belonging have a huge impact on happiness. The results of our investigation, i.e. the strong influence of religious attendance on happiness suggests, appear to corroborate that. Further study focusing on the impact of religion and other community focussed bodies would be an interesting direction to go. It would be good to disentangle the loose factors we are conjecturing are at play here - at its core how much of this impact is due to the associated community and how much faith itself. A possible way to differentiate these two things would be a comparative study with diverse religious groups on one side to isolate the faith angle, and non faith based institutions which exhibit these characteristics e.g. sports groups, hobby groups and

other institutions with a common goal. This is an encouraging direction for further research into the impact of social support networks and community involvement on happiness.

Furthermore, better performances of models in predicting unhappiness rather than happiness could suggest that negative experiences or deficiencies may have simpler patterns that can be captured by socio-economic variables. Thus, the comprehensive analysis of socio-economic factors influencing happiness, as described in the report, aligns with global studies on happiness, such as the World Happiness Report (WHR, https://worldhappiness.report/ed/2023/). The WHR highlights the importance of social support, income, life expectancy, freedom, trust and generosity as determinants of happiness. The predictive modelling and trait importance analysis in our study are consistent with these findings, focusing on similar factors such as income, work prestige and education.

In the context of Finland and the other Nordic countries, which often rank at the top of the happiness league table, the results of our study on predictors of happiness are particularly pertinent. These countries' high happiness scores can be linked to strong social support systems, high levels of trust and comprehensive social protection policies, which our analysis also identifies as key determinants of happiness. The emphasis on social and economic support in policy-making, as in Finland, is reflected in our study's identification of socio-economic factors as significant predictors of happiness.