Abstract

This report explores the predictions of the 2019 World Happiness Report dataset through two different machine learning models – linear regression and deep learning. Their performance in predicting the happiness score based on the features of the dataset such as GDP, Social Support and Generosity. The final outcomes are that both models perform the predictions very accurately due to careful EDA, data preprocessing and feature selection.

Background and Problem addressed

The problem I wish to address is to figure out the most significant factors in people living happier lives and to be able to predict the happiness of different countries so that every country and every year of collected data can be considered. Therefore, the solution of this problem should allow me to differentiate between the dystopian countries (lowest happiness scoring countries) from the utopian countries (highest happiness scoring countries), compare the overall happiness as well as the values of the factors that contribute to that happiness whilst also portraying the varied significance of the factors for each specific country.

For this problem, my chosen dataset is the World Happiness Report for 2019 (available via Kaggle), which provides the overall happiness ranking and scoring of 156 different countries throughout the world. The rank and score are dictated by the factors of a country: economy, social support, life expectancy, generosity and government corruption. This data has been gathered by the Gallup World Poll, a global survey that has been conducted for over 80 years and is so vast that it currently reaches 99% of the adult world population to obtain people's will: hopes, dreams, behaviours as well as their happiness.

Whilst I am fairly certain that the data source is legitimate and hopeful that the data collected is unbiased, I must acknowledge that there is a possibility of some bias (bogusly boosting or undermining of certain countries' ratings). After some research of the credibility of the Gallup World Poll, the Media Bias Fact Check resource website concludes that as a source, the Gallup World Poll has 'minimal bias' and rates them 'mostly factual' with a '69% accuracy rating'. As a result, whilst the source of the bias check may also have their own bias, I am more confident in the data source's credibility.

Exploratory Data Analysis

Dataset description, visualization with supporting figures and tables.

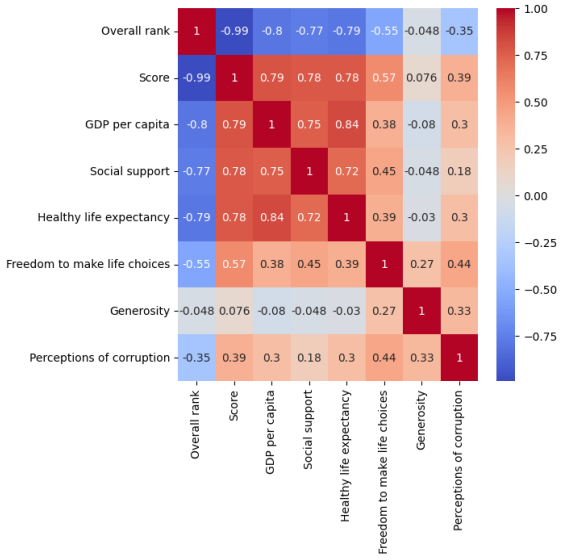
The Exploratory Data Analysis (EDA) process is vital as it allows us to gain understanding of the data and become familiar with the dataset; its structure, size key features and data types. This stage will be beneficial for data science (in this case machine learning) as an abundance of meaningful information can be discovered, such as patterns, relationship and whether there are any outliers. These insights will lead to prevention of mistakes, suitable feature selection, cleaning of the data (removing non-essential features, handling human error and identifying outliers) and aid in choosing the right method.

After importing the dataset, its features and values can be quickly seen via 'data'. The first feature is rank – every country in this dataset is ranked by their happiness score, with 1 being the happiest country and 156 being the least. Country is the name of the country as a string. Score is the happiness score of a country and, according to the dataset provider, the other factors 'describe the extent to which these factors contribute in evaluating the happiness in each country'. The factors contributing to the happiness score are; GDP per capita, social support, healthy life expectancy, freedom to make life choices, generosity and perceptions of corruption present in the country.

Through importing the relevant libraries, I explore the dataset: via 'data.describe()' - the output in which gives the feature names and some information about each feature's values – very useful for understanding the typical values and the spread of data. The columns tell us the; count, average value, standard deviation, the minimum and maximum values and the 25%, 50%, 75% percentiles of data which tells us the distribution of data.

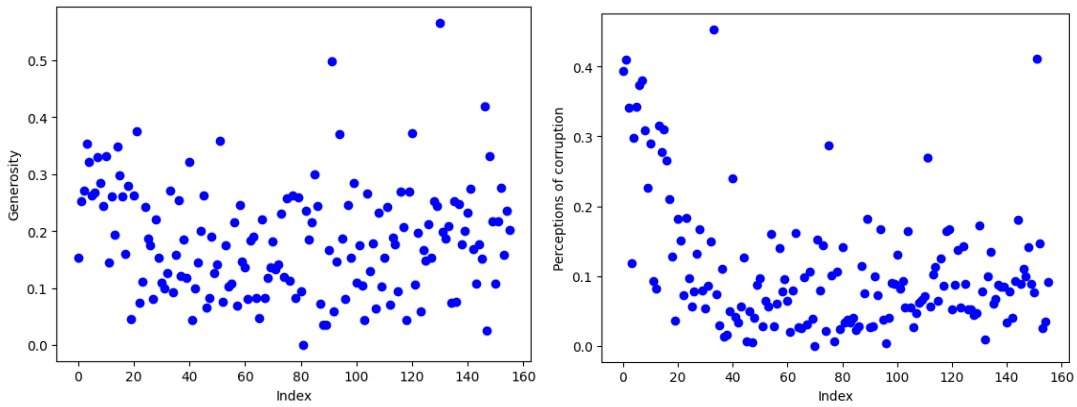
Through 'data.info()' I check the data types. Then I double check for any null values via 'data.isnull()sum', getting the sum of null values in each feature.

To see the relationship between the features, a model depicting the linear correlation between variables would be most suitable:



In the correlation visualisation, I use a heat map with the cool warm colour map to portray the values of the correlations - the negative (blue, -1), positive (red, 1) and neutral (white, 0) values. As a result, the extremes in this model are easily identifiable, there is a clear midpoint, it is distinct visually as well as being intuitive with cold blues and warm reds which we naturally associate with negative and positive.
A positive correlation means as one variable increases, so does the other and a negative value means as one variable increases, the other decreases.
The most negative correlation is with the rank and score (0.99) - this is expected as the lower the rank, the higher the happiness.
The strongest positive correlations with score are GDP (0.79), social support (0.78) and life expectancy (0.78) which suggests happiness is mostly associated with higher values in those factors.
The freedom and corruption (0.57, 0.39) of a country has some positive correlation with the score, but not as much as the previous features.
The weakest correlation with score appears to be generosity (0.079), suggesting that it barely affects the happiness score when its value changes.

To search for outliers in the data, a scatterplot of every feature in the data where values are plotted as scatter points can help visualise the patterns, trends and outliers. In the scatterplot model below, most of the features have a pattern of data without any isolated points, however the visualisations for the generosity and corruption features highlight some outliers. Whilst the number and level of abnormality (not too far from rest of the data points) is low, these outliers are still important to consider during machine learning as their performance may be reduced due to those outliers.

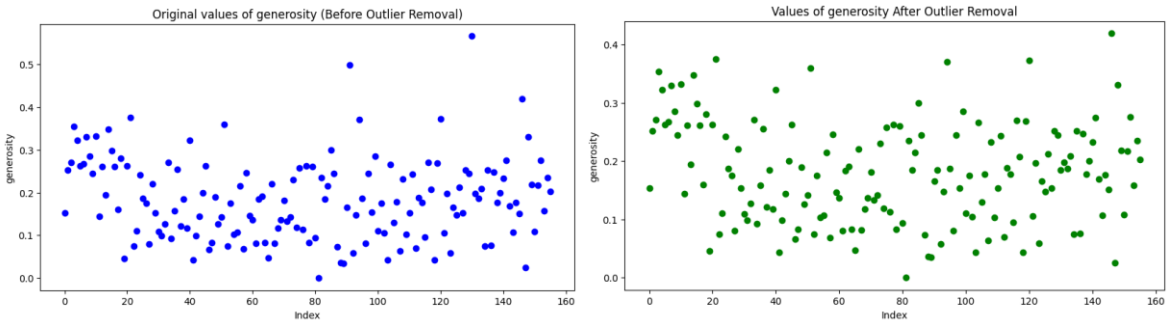


Data Pre-processing and Feature Selection

The Data Preprocessing stage is very important, and generally more important than the model architecture itself, as it prepares the data for the machine learning models – unprepared data will result in very poor performance of the models as the data taught is incorrect through specific pre-processing methods.

To start this stage, renaming the names of the columns to names that are shortened down and simplified will not only be beneficial in writing code but also in visualisations as labels for axis will be shorter. Another benefit would make names of columns easier to understand. A couple examples include ‘GDP per capita’ being renamed to ‘GDP’ and ‘Freedom to make life choices’ to simply ‘Freedom’

As outliers have been spotted in the EDA stage previous, a function to remove them from the data would be essential if they prove to affect the performance of the machine learning models. One method includes using the interquartile range and then replacing the found outliers with the mean of the column. Below you can see the before and after of the outlier removal process.



Machine learning models react sensitively to the scale of the data – wider ranges of data which outliers cause as they have the lowest/highest values results in a wider scale of data, reducing model performance as the learning process becomes less stable and slower.

Feature selection is another important aspect to consider as deciding on which features will be used to train the machine learning models is very significant and can improve the model performance, prevent overfitting and reduce complexity. In consideration of feature selection – removing the features with the least influence on the score would reduce noise and data redundancy which would improve the performance of the model. Reducing multicollinearity for regression models is very beneficial and this can be done by dropping some features with the highest influence.

Machine Learning Models

Machine Learning Model 1: Linear Regression

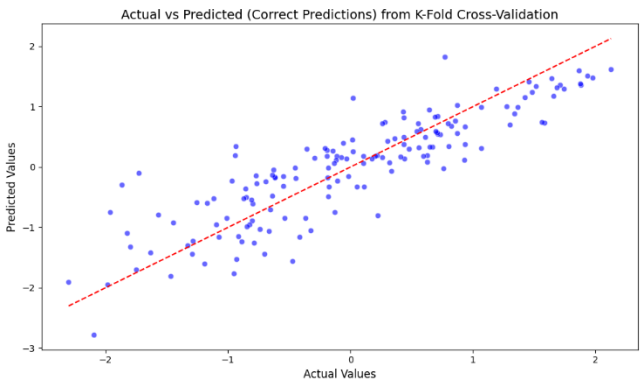
Summary of the approach: Regression is a suitable choice for this dataset as the data has a linear relationship between the score and independent variables – the features like GDP, life expectancy and the target variable are continuous which fits for linear regression. The model would be suitable as it is very interpretable as the model coefficients tell how direct the relationship between every feature and the score, allowing for understanding how important each feature is in dictating the score value. After deciding on regression, one must question - linear or logistic regression? According to the article written by Horacio Matias Castro and Juliana Carvalho Ferreira, a continuous output variables (such as ‘score’ and ‘gdp’ in this case) is used for linear regression but if the dataset had a categorial output variable such as ‘happiness’ where values can either be one or the other (happy or not), then logistic regression is the choice. The model will be tested on its overfitting and generalization by splitting the data either into train/test splits or via k-fold cross validation, both of which would be applied during the training of the model and their evaluations of their respective linear regression model after training will be compared. Model training: The data is prepared - ‘train_x’ being the independent variable and ‘train_y’ being the target variable that will be predicted. The train/test splits will employ a 10/90 split which leaves 10% of the data for testing and 90% of the data for training. This is most suitable as it leaves most of the limited data available (due to the small sized dataset) for the model to learn the patterns, allowing for better learning rate and variance reduction. 10% gives only

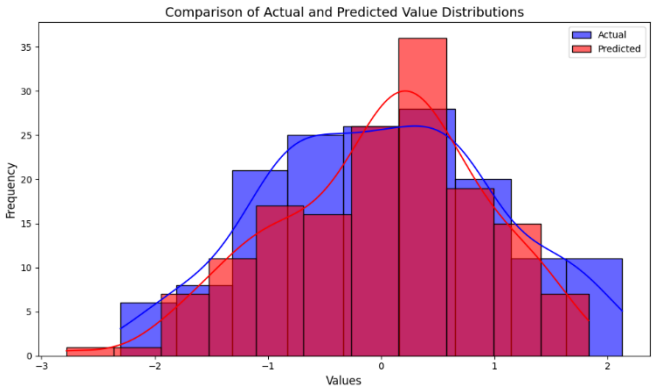
16 data points to to evaluate the predictions with but it is enough in this case. The linear regression model is created and then fitted, where the coefficients and the intercept of the data is calculated to figure out the best fitting line which minimizes the difference in predicted values and actual values. Using K-Folds cross validation rather than just the train/test split is a method to consider.

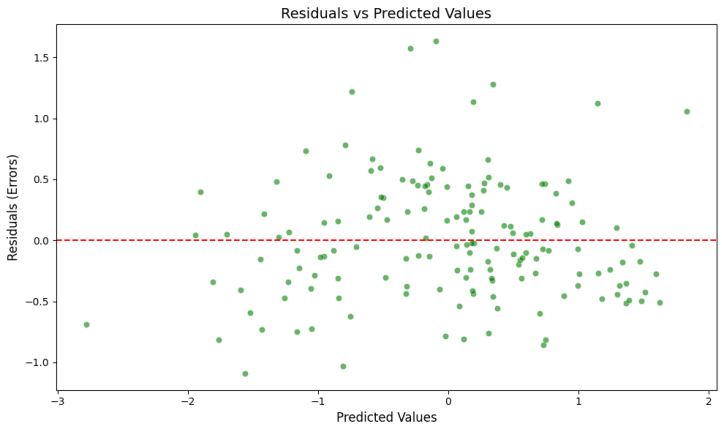
Evaluation: The best model explains 75.89% of the variance and has an R-square score of 0.76 which means the model is fitting the data quite well and explains a good amount of the variance in the data, therefore the model has clearly captured the most important relationships between features and the score. The Mean Absolute Error is 0.39 which is quite small compared to the scale of the data so this means the prediction accuracy is also quite high.

Results: Scatter Plot: actual values versus predicted values where the regression line is also plotted and overall the predictions are fairly close to the line. There are, however, a few deviations for the extreme values. Histogram: alternative visual of actual versus predicted. Can see more clearly that the predictions line up close to the actual values as the data is distributed similarly, despite the predictions being less dispersed which suggests underrepresentation of variability.

Residual plot: to see if difference in prediction and actual values and randomly distributed. Can see that data does not follow any pattern and the data is distributed randomly which is a great indicator that the model is indeed doing well.







Machine Learning Model 2: TensorFlow Neural Network, Deep Learning

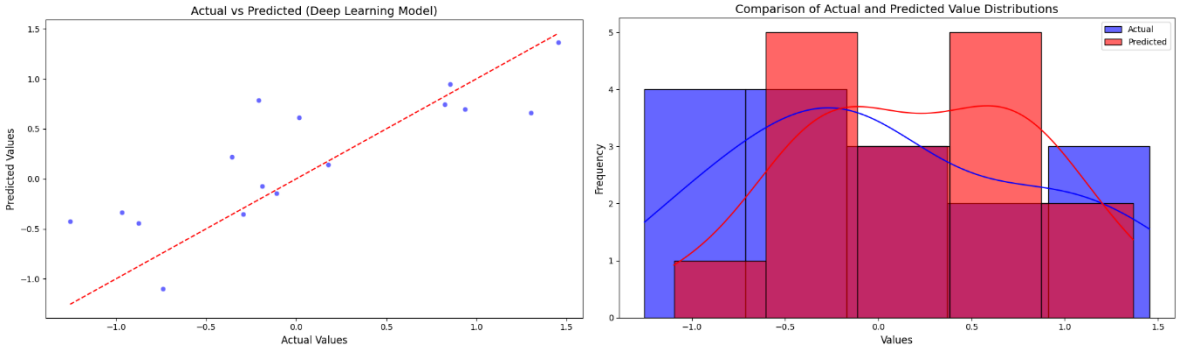
Summary of the approach: According to Laith Alzubaidi's article on neural networks, neural networks, especially convolutional neural networks, automatically extract meaningful features from raw data which

removes the need for extensive human-driven feature engineering. For this case, a deep learning model can be used to predict the score based on the rest of the features in the dataset. The architecture of this neural network would consist of three layers – one input layer to accept features, two hidden layers with 64 and 32 neurons, and one output layer to predict the score. The two hidden layers in the model will allow for learning of the non-linear and complex relationships between the features and the score where the output from the output layer is continuous which is suitable in this case for linear regression. Parameters include once again the train/test split, 50 epocs (50 iterations over training), batch_size of 32 (number of samples before weight update) and the validation split of 0.2 (20% of training data for validating performance on unseen data.This is a much more complex method is suitable as it will be able to capture the more complex relationship between features and the score which should output with a more accurate prediction compared to the linear regression model that may miss these relationships. The model will then be evaluated via the same metrics.

Evaluation: The best version of the model explains 69.43% of the variance and has an R-square score of 0.65 which means the model is fitting the data quite well and can explain for most of the variance in the data, therefore the model has clearly captured the most important relationships between features and the score. The Mean Absolute Error is 0.36 which is quite small compared to the scale of the data so this means the prediction accuracy is also very high.

Results:

Scatter plot: Can see that the predictions have high accuracy and line up to the regression line. Histogram: can see that predictions line up very well other than one dip in actual values that is not predicted.



Results comparison across the models built

Linear Regression

First model has good test performance (0.37 MAE), but this was using the data after outlier removal. I experimented by using the same model but with the uncleaned data where outliers were present, and the performance increased somewhat (0.35 MAE, 0.02 improvement). I then tried the K-Folds Cross Validation method as the random state of the train/test split data would affect the performance very significantly. One state would provide a good result of 0.35 MAE but another performs with 0.5 MAE, which is not consistent at all and suggests the model is missing out on the non-linearity of the data and that varied data would result in different model performances. Via K-Folds, performance yielded a MAE of 0.39, 75.89% variance, 0.76 R-Squared which was lower than some results from the previous model, but this was much more consistent as it did not change depending on the random state of data. This validation also had a higher r-squared score compared to the train/test splits – 0.76 compared to 0.71, but a very slightly lower explained variance of 75.89% compared to 76.16%.

Deep Learning Neural Network

First model had great test performance (0.34 MAE, 81.41% variance, 0.71 R-Squared) and this was for the data that was scaled and cleaned. For the second model, the feature selection was enhanced by removing the 'generosity' and 'social' features. Generosity contributes the least whilst adding noise to the model which can reduce the precision of predictions. Removal of such a feature improves generalization and reduces overfitting. Social is one of the most influential features and in being so is most likely highly correlated with the other most influential features like GDP which means there is multicollinearity – the model's performance will likely suffer due to this as it will be more difficult to differentiate the contributions of each feature which will cause unnecessary complexity in the deep learning model. This model had the best test performance (0.31 MAE, 78.17 variance, 0.76 R-Squared) with a lower MAE despite a decrease in explained variance. I tried to improve the performance further by experimeting with the parameters. In reduction of batch size and the validation split, which may help for smaller datasets such as this case, the model suffered slight performance loss. For the final model, the model architecture was made more complex through implmentation of two batch normalization layers and early stopping. Batch normalization normalizes the input features when trianing process and ensures the model is less sensitive to the weights and learning rate of the model. Early stopping is a method that halts the training process when epochs noticeable reduce in model improvement, preventing overfitting. These two methods, however, reduced the model's performance significantly (0.49 MAE, 50.32 % explained variance, 0.46 R-Squared).

Conclusion, recommendations, and future work

In conclusion, both the linear regression and deep learning models clearly present the ability to predict the score variable accurately whilst explaining the majority in the variance of the data, as well as having high R-Squared scores. The deep learning model has a lower MAE and explains more of the variance so it had the more precise predictions compared to the linear regression model. The choice between them comes down to either going for interpretability of the linear regression, or the predictive performance of the deep learning.

I would like to learn about the additional deep learning model architecture layers I could implement – the additional layers of batch normalization did not benefit the model whatsoever, but nonetheless I am quite happy with the results of both models as they are not only suitable for the dataset (despite it's very small scale which was difficult to manage), but in their accuracy of their predictions being high.

References

World Happiness Report dataset: https://www.kaggle.com/datasets/unsdsn/world-happiness/data?select=2019.csv

Gallup World Poll's statement: https://www.gallup.com/analytics/318875/global-research.aspx

Media Bias Fact Check: report of the Gallup World Poll: https://mediabiasfactcheck.com/gallup/

Horacio Matias Castro, Juliana Carvalho Ferreira - Article on Linear Regression Models: https://pmc.ncbi.nlm.nih.gov/articles/PMC9747134/

Laith Alzubaidi, others – Review of deep learning: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8