What makes people in a country happy?

The World Happiness Report is a landmark survey of the state of global happiness that ranks countries by how happy their citizens perceive themselves to be. We have the data sets of the happiness score along with other features of the countries such as GPD per capita, Social support, Healthy life expectancy, freedom to make life choices, etc. Our goal is to gain insight into the state of happiness in the world today.

The first step for a data science project is to gather the data, but the data is already collected and we can skip this step. The second step is to pre-process the data and then we can do the EDA(exploratory data analysis and visualization).

I first check to see if the data has any null values in the CSV by using the 'df.isnull().sum()' method which checks the number of the null values for all the years from 2015-2019. There is only one null in the year 2018 and I used the interpolation method to estimate the missing value in the year 2018. The interpolation method is one of the imputation methods that calculate the missing value by predicting it using linear regression. I chose to use this method instead of other methods such as mean imputation, random imputation because interpolation has the least effect on the overall dataset and it has a good prediction model to support it.

I then calculated the central tendencies of the happiness score over the years by using the mean() and median() methods. I decided to plot the score for each year to have a good visualization to see if the happiness score increased or decreased over the years. Based on the plot, the mean fluctuated over the years and the mean value in 2015 and 2018 is almost the same with a score of around 5.37, meaning that the mean did not change much over the years. However, the median did have an increasing trend with a 5.23 in the year 2015 and 5.38 in the year 2018.

I also checked to see which countries have stable ranking over the years, and which countries improved their rankings by merging the years from 2015 to 2018 based on the

‘Country’ column. I calculated the change by summing up the absolute value of the difference between the year and its subsequent year. Whichever country has the smallest change in rank have the stable ranking. The Netherlands and New Zealand have the most stable rank with a change value only being 1. Then Iceland, Canada, Australia also have a stable rank with a change value being 2. To determine which countries improved their ranking, I calculated the difference of rank between the year 2015 and the year 2018 and find the countries that have a difference bigger than 0, meaning that the rank in 2018 is smaller than the rank in 2015 and that is the increase of ranking. There are about 81 countries improved their ranking with a maximum improved rank of 44 (Ivory Coast).

After that, I made some plots to visualize the relationship between happiness score and other features. I used a seaborn method called regplot to make a scatterplot and the best fit line with independent variables being all the features and target variables being ‘happiness score’, and I provided the correlation coefficient so I can compare them later. The correlation between the happiness score and ‘GDP’ is 0.8, for ‘social support’ is around 0.75, for ‘healthy life expectancy’ is around 0.78, for ‘freedom to make life choices’ is 0.54, for ‘Generosity’ is 0.14, for ‘perceptions of corruption’ is around 0.41. We found that ‘GDP’, ‘Social support’, and ‘healthy life expectancy’ are the most important or most correlated features to have a good happiness score. All these features contribute to happiness with some weights more and some weights less. If I were a president of a country, I will consider the cost to improve certain features and the correlation of the features and make the best out of my resources. If the cost to improve any of the features are the same, I spend most of my resources on ‘GDP’, ‘Social support’, and ‘healthy life expectancy’ since they have the highest correlation coefficient, and I will focus on the rest features if I still have leftover resources.

The third step for data science project, which is ‘analysis and modelling’ and I made three models for this part. The first model I used is Linear Regression, also known as Ordinary least squares Linear Regression. This model is very common for linear datasets. It firstly calculates the square of distance between actual data from test data and predicted data from linear approximation, then it finds the smallest sum of the square of distance by fitting coefficients K1, K2, K3…Kn, n is the number of variables in the dataset. I called fit() function on 2018 data to train the model, then called predict() function to predict the 2019 data based on variables of 2019. And the mean squared error for linear regression on this dataset is 0.276.

I also used Random Forest Regressor. It firstly uses different subsets of the training dataset to fit some classifying decision trees, then it uses averaging to improve predictive performance and also handles over-fitting cases. I set the maximum depth of the tree to be 8 and random state parameter to be 1. Similar to linear regression, I called fit() function on 2018 data to train the random forest regressor, then called predict() function to predict the 2019 data based on variables of 2019. And the mean squared error for linear regression on this dataset is around 0.24, which is better than linear regression.

The third model I used is K Neighbors Regressor. It predicts target by local interpolation of the testing data and the nearest neighbor in the training data. Similar as the two models above, I called fit() function on 2018 data to train the random forest regressor, then called predict() function to predict the 2019 data based on variables of 2019. The mean squared error is 0.16, which is the best among all of them.

For my own model, I used the linear function approach, which is the format of y = ax + bx + cx + dx + … + c. For coefficients, I used the one found in linear regressor and same for intercept. The mean squared error for my model is 0.27, which is not bad.