Predicting Happiness Scores in the World Happiness Report

Using linear regression, polynomial regression and huber regression

Section 1: Introduction

In this project, I want to build a model that predicts the happiness score of a place in the World Happiness Report (WHR), based on local statistics including pollution index and local purchasing power index. This model could help us predict the happiness score of a particular population, even though it is not studied in the WHR. The formulation of this as a machine learning problem will be presented in-depth in section 2.

This model could for example help individuals predict their expected level of happiness before moving to a certain place, or to simply help whoever is interested to understand the level of happiness of a certain population.

Since the output of the model (the happiness score) is numeric, I concluded that this is a regression problem. I trained the model with three different regression models: linear regression, polynomial regression (with degree 2) and huber regression. The training process of these models is discussed in detail in section 3, and the results is presented in section 4. In section 5, I conclude the project and propose potential improvements.

Section 2: Problem Formulation

This application can be modelled as an ML problem as shown below:

Data points: Countries
Labels: Happiness scores of each country in the World Happiness Report 2020
Features: Countries' pollution index and local purchasing power index recorded in 2020

The labels (happiness scores in the WHR) is hard to obtain, since the study is based on aspects that are very hard to measure or quantify, including freedom to make life choices and generosity, among other things. On top of that, WHR does not cover all places on earth, and therefore the happiness score of some places might be impossible to obtain. However, the features (pollution index and local purchasing power) can easily be obtained from authorities or from the internet (e.g. Numbeo or Wikipedia).

Section 3: Method

Source of data. Results of the World Happiness Report 2020 (to be used as labels) was obtained from WHR's official website, local purchasing power dataset was from this Kaggle page and pollution index dataset was from this Numbeo page.

Feature selection. After studying multiple sets of data, I chose the pollution index and local purchasing power index as the features for three reasons. Firstly, these features show correlation with the labels (see fig 1 and 2), and therefore prove that they are relevant features. Secondly, these two features do not show correlation from each other, therefore their contribution to the prediction is maximised (if I were to use GDP and local purchasing power as features, their joint contribution to the prediction will be smaller, because they are related to each other). Thirdly, these features are easy to obtain from authorities or from the Internet, therefore increases the utility of the model.

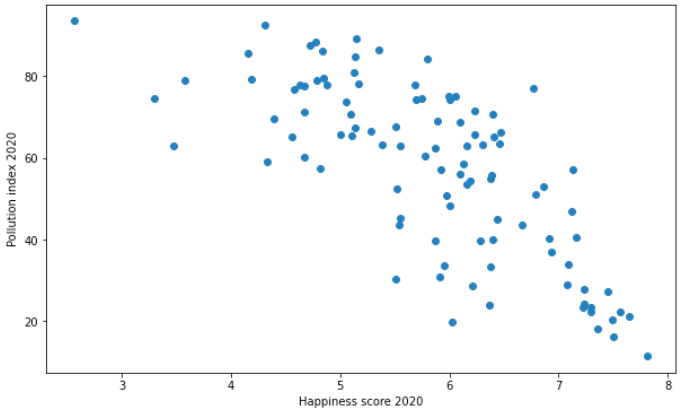

Fig 1. Scatter plot showing correlation between pollution index and happiness score

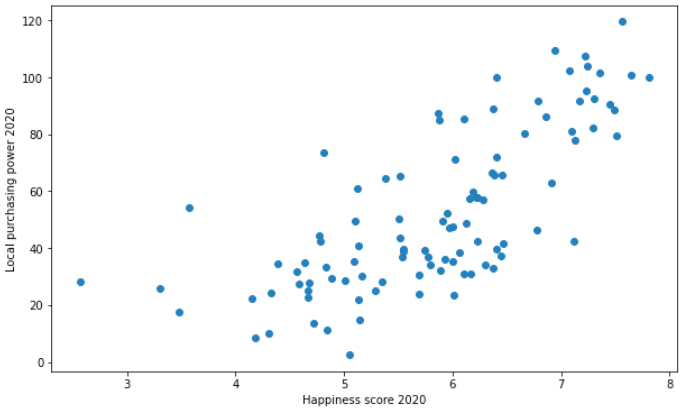

Fig 2. Scatter plot showing correlation between local purchasing power index and happiness score

Handling missing data. To simplify the training steps, I eliminated data points (countries) with any missing features or the labels, which left me with 101 data points in the dataset. It is not a particularly big dataset, but it was a big enough to allows me to obtain reasonable training results. I believe it is also a representable dataset, considering that it is more than half of the total number of countries we have in the world.

Hypothesis space and loss function. I observed a linear relationship between the features and the label (see fig 1 and 2), therefore I believed a linear hypothesis (linear regression) would be a reasonable choice. In addition, I also experimented with a polynomial regression with degree 2 to find out if it would fit the data better. On top of that, since there seems to be some noice and potential outliers in the data, I also experimented with huber regression which is a robust model toward outliers (Jung, 2021).

For loss function, I chose the mean squared error loss to compare the performance (training, validation and testing error) of these models. This loss function works well with machine learning problems involving numeric labels (Jung, 2021), which in this case are the happiness scores.

I used Scikit-learn's ready-made classes `LinearRegression`, `PolynomialFeatures` and `HuberRegressor` for training, and the `mean_squared_error` class to obtain the loss / error of each model.

Model validation and testing. I used the k-fold method with `k = 5` for model validation, which means the data is repeatedly split into training and validation set k times (Jung, 2021). This allows me to make use of all the data for both training and validation. This is done with the off-the-shelf `KFold` class from the Scikit-learn library.

My training dataset had 101 data points, which was slightly too small to be split into training and testing sets. Instead of doing a train-test split, I decided to use data of 2015 for testing. The reason why I chose data of 2015 is that the happiness scores (the label) and the indices (the features) have changed from 2015 to 2020, which avoid the scenario where the testing data is very similar to the training data (which might be the case if I choose data of 2019 for testing). If the model - trained with data of year 2020 - achieve good prediction result on the data of 2015, it should be a stronger indicator of its accuracy.

I collected the local purchasing power index and pollution index recorded in 2015 from Numbeo, and the result of WHR 2015 from the WHR website. After combining these data, I got a testing set of 84 data points.

Section 4: Results

	Linear regression	**Polynomial regression (degree 2)**	**Huber regression**
Average training error among 5 splits	0.4184	0.3922	0.4218
Average validation error among 5 splits	0.4544	0.4445	0.4441
Testing error	0.5352	0.8711	0.5534

Table 1: Training, validation and testing errors

Table 1 shows that the validation errors are very similar among the three models. However, linear regression has the lowest testing error, therefore this is my final choice of model. Polynomial regression (with degree 2) shows signs of overfitting, as its training error is the lowest while its testing error is the highest among the three. Huber regression does not seem to show significant improvement comparing to linear regression, which implies that the noise/outliers in the data was not significant.

Since pollution and purchasing power indices are locally measured, I think the model can be uses to predict the happiness scores of not only countries, but also cities and towns. Some prediction examples are shown in table 2.

	Tampere (in Finland)	Kuala Lumpur (in Malaysia)	Antwerp (in Belgium)
Pollution index (from Numbeo)	15.17	66.79	59.95
Local purchasing power index (from Numbeo)	103.76	67.27	88.7
Predicted happiness score (linear regression)	7.44	5.91	6.53

Table 2: Some prediction examples of the trained model

Section 5: Conclusion

In this project, I studied three different regression models to learn a happiness score prediction model. The goal is to train a model that can predict the happiness score of a population based on the place's pollution index and local purchasing index, without the place being studied by the World Happiness Report.

The model was trained with data of the year 2020, and tested with data of the year 2015. The k-fold method was used for cross validation. Even though the validation errors did not play a significant role in the final model selection, it validated the model's expected behaviour during the training process.

Linear regression is my final choice of model because it has the best performance with the testing set (with testing error being the lowest). Since the happiness score ranges from 0 to 10, I think its testing error 0.5352 is acceptable. However, there is no benchmark level available to validate the level of the model's performance.

To increase the accuracy of the model, perhaps one to two more features can be gathered to train the model, due to the multi-faceted nature of a population's happiness. Some examples are crime index, healthcare index and average life expectancy, which can all be easily collected. However, more features might lead to overfitting due to limited data points, and might call for a principal component analysis.

Reference

JUNG, A., 2021. *Machine Learning. The Basics*. [view 19 March 2020]. Available from: mlbook.cs.aalto.fi