To What Extent Can Regression Techniques in Predictive Analytics Be Used to Estimate the Happiness Levels of Countries Given a Certain Set of Parameters?

Subject: Computer Science

Word Count: 3983

Table of Contents:

Introduction

The use of algorithms to evaluate future events is more prevalent than ever. Using machine learning, computer scientists apply the algorithms in real life use cases. A subdivision of this is predictive analytics. Used in a variety of fields such as credit scoring, fraud detection, supply chain management, and human resource management, predictive analytics is a technique used to mitigate risk (Halton).

However, is predictive analytics only applicable in fields of business and human management? The aim of this paper is to use predictive analytics to understand human behaviour on a mass scale, specifically the use cases in predicting happiness levels of countries all over the world from certain factors using the World Happiness Report. The specific predictive analytics models this paper will be using are the linear regression and the regression tree (a subcategory of decision tree) models.

What is Predictive Analytics

To first understand what predictive analytics is, a brief understanding of machine learning as a whole is required. Machine learning is “branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy” (IBM). Predictive analytics is a branch of machine learning, which “makes predictions about future outcomes using historical data combined with statistical modelling, data mining techniques and machine learning” (IBM).

The predictive analytics models are mainly split into two types: supervised and unsupervised learning. Supervised learning is when the models “train on sample data that specifies both the algorithm's input and output,” whereas unsupervised learning is when the model learns by making connections without assistance or knowing the output (Amazon).

Supervised learning, which will be the focus of this paper, is further classified into two different types of models: classification and regression models. The former “ predict(s) the membership of values to (a) certain class,” such as if a person owns a car, while the latter “predict(s) a number”, such as the

price of a house, given certain factors (Kumar). Because this paper is predicting values and not sorting values, regression models will be the main focus.

Regression Models

Regression models are the most simple and popular models in predictive analytics. Regression models study the relationship between one dependent variable and one or more independent variables, specifically how specific variables, or factors, influence the change in another variable within large data sets (Kumar). In a regression model, "the dependent variable Y must be continuous, while the independent variables (X) may be either continuous (age), binary (sex), or categorical (social status)" (Schneider).

Linear Regression

Linear regression is applied to a data set when the relationship between the dependent variable and independent variable increases/decreases by a constant value (linear). Linear regression is "a global model," meaning that there is one formula that is used for the whole predicting. Because of this, when the linear model has multiple features, a formula can be very difficult to construct and decipher for a human (Carnegie). Because of this, machine learning can greatly simplify the process.

Linear regression with one feature is modelled by the equation:

$$y = mx + b + \epsilon$$

where y is the dependent variable, x is the independent variable, m is the slope of the linear line, b is the y-intercept, and ϵ is the random error.

To model multivariable linear regression, the mx value is increased by the amount of features, each mx representing a different feature. Since the data set analysing will have multiple independent variables, the use of the multivariable regression will be necessary.

Logistic Regression

Logistic regression is used when the "dependent variable is discrete," such as a pass/fail. Although it sounds like a classification algorithm, it is very related to linear regression, as it uses the linear regression algorithm but applies the sigmoid function to it. The sigmoid function is:

$$f(x) = \frac{1}{1 + e^{-Y}}$$

Therefore, the logistic regression formula becomes

$$Y = \frac{1}{1 + e^{-(mx + b + \epsilon)}}$$

However, since the values in the data set are continuous and not discrete, the logistic regression formula does not apply.

Loss/Cost Function

Since there are infinite possibilities when trying to find the line of best fit, a brute force calculation will be inefficient (as it will require immense computing power and time), if not impossible (since the computation can theoretically last forever). With the case of the linear regression algorithm, a cost and loss function is used. A loss function is "the error for a single training example" (Shah). The further away the estimate is from the real value (the higher the error) the higher the output of the loss function. The cost function is the "Cost function is the sum of losses from each data point calculated with loss function" (Luo). For linear regression, the most commonly used cost function is the root mean squared error, which is modelled by the equation,

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y} - y)^2}{n}}$$

,where y is the actual value, $\widehat{y}$ is the predicted value, i is the starting value, and n is the number of data points (Luo). It is notable that this cost function is a parabola with one global minima and no local minimas or maximas, so the optimal solution of the cost function is at the vertex, where the difference between the predicted value and actual value is minimised. To find the vertex, there are many methods. but in this paper, two methods will be discussed: the gradient descent method and the normal equation method.

Loss Function for Linear Regression: Gradient Descent

Gradient descent is a algorithm that tries to find the point on the parabola where the gradient/derivative of the cost function is 0, ie. the vertex. It does this by choosing a point on the graph, calculating the gradient at that point, and moves left or right accordingly to minimise the gradient. Essentially, the algorithm repeats the equation below until it has reached the closest point to the vertex, meaning that it is an iterative algorithm. The equation is shown below as

$$\theta' = \theta - \alpha \cdot \frac{\partial cost}{\partial \theta}$$

where θ' is the new theta (point on parabola), θ is the old point, α is a constant step that is predetermined, and $\frac{\partial cost}{\partial \theta}$ is the derivative of the cost parabola in respect to theta (the gradient of the point). If the point is to the left of the minimum, $\frac{\partial cost}{\partial \theta}$ will be negative, so θ' will be to the right of θ. If the point is to the right of the minimum, $\frac{\partial cost}{\partial \theta}$ will be positive, so θ' will be to the left of θ. Note that if the predetermined α is too large of a value, the formula will continue to go left and right of the minimum without ever getting close to it. If α is too small, the formula would take much more time than needed (Khan). In theory, if given infinite time, it can reach the vertex, minimising the error as much as possible. However, in reality, due to the constraints of the α value, it can get close to the vertex but never reach it.

Loss Function for Linear Regression: Normal Equation

As mentioned before, gradient descent is an iterative process. However, this is only practical through the use of computers. If one were to use gradient descent to find the vertex without computers, it would be time consuming depending on the predetermined step.

In contrast, the normal equation is analytical, meaning that it computes the lowest cost without parameters like the learning rate or the number of iterations. It does this through matrix multiplication, specifically by grouping the independent variables into one matrix and the output in another matrix.

The normal equation is presented below with the equation

$$\theta = (X^T X)^{-1} . (X^T y)$$

where θ is the point closest to the vertex of the cost function, X is the input feature of each variable, and y is the output of each variable (geeksforgeeks). The normal equation method is best for data sets that are relatively small, in contrast to gradient descent.

Decision Trees

Like a linear regression model, a decision tree is used to "create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features" (scikit). There are two types of decision trees: a classification tree and a regression tree. A classification tree deals with data sets of qualitative data, such as if a group of people have or do not have a house (this type of decision tree is categorised in the classification type of model). A regression tree deals with data sets of quantitative data, such as the cost of various houses (this type of decision tree is categorised in the regression type of model). For the purpose of this paper, a regression tree will be used.

In a decision tree, there are a few terms to explain the way it performs the learning process. First is the node, which refers to the entire dataset. The node is separated into multiple sub nodes using a

condition. In the case of a regression tree, it splits the node by comparing it to a certain value. Once the node cannot be split any further into branches, it is called a leaf node. In essence, it is similar to a binary tree in structure (IBM).

The difference between the linear regression model and the regression tree model is that the former constructs the line that has the least error out of every possible line constructable within the data set, whereas the latter "learn(s) by splitting the training examples in a way such that the sum of squared residuals is minimised. It then predicts the output value by taking the average of all of the examples that fall into a certain leaf on the decision tree and using that as the output prediction." (Dan).

Note that there is another technique called Random Forests, which uses multiple decision trees to increase accuracy. However, the data set being used is not large enough to be a feasible option in this case.

Loss Function for Decision Trees: Recursive Binary Splitting

To minimise the cost function used before, decision trees use a different method compared to the ones of linear regression. Regression decision trees use a "top down greedy process called recursive binary splitting." The greedy aspect is how when it makes a split, the algorithm does not consider minimising the cost function of subsequent splits, only focusing on the current split. It repeats the process until a "stopping criterion" is satisfied, where splitting further does not decrease the cost function significantly (Foley).

However, running through every possibility of a tree is time consuming, so regression decision trees use cost-complexity pruning, which quantifies the tradeoff between error and tree size to limit the possible trees the algorithm has to run through. This is modelled by the equation

$$R_{c_p}(T) = R(T) + c_p|T|$$

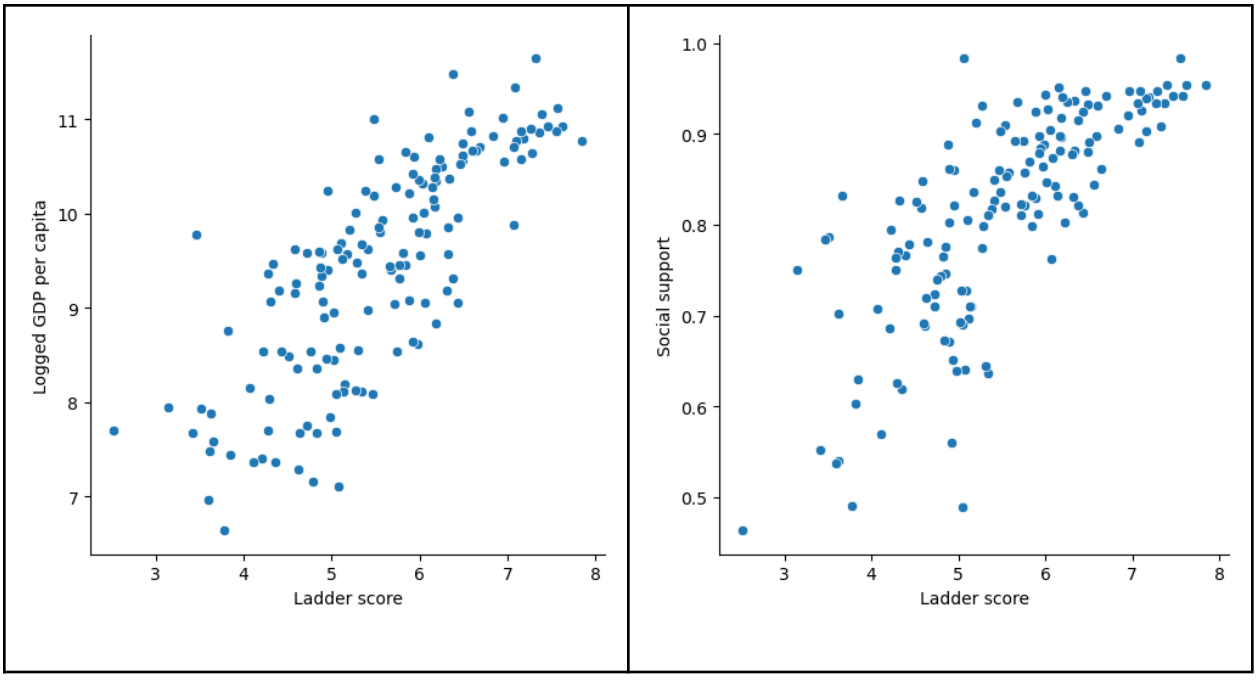
where $R_{cp}(T)$ is the sum of the risk, c_p is a cost-complexity factor, and $|T|$ is a multiple of the tree size. c_p can theoretically be any value from 0 to infinity, but there are only a finite number of "optimal values," which the decision trees use cross-validation to find (Foley).

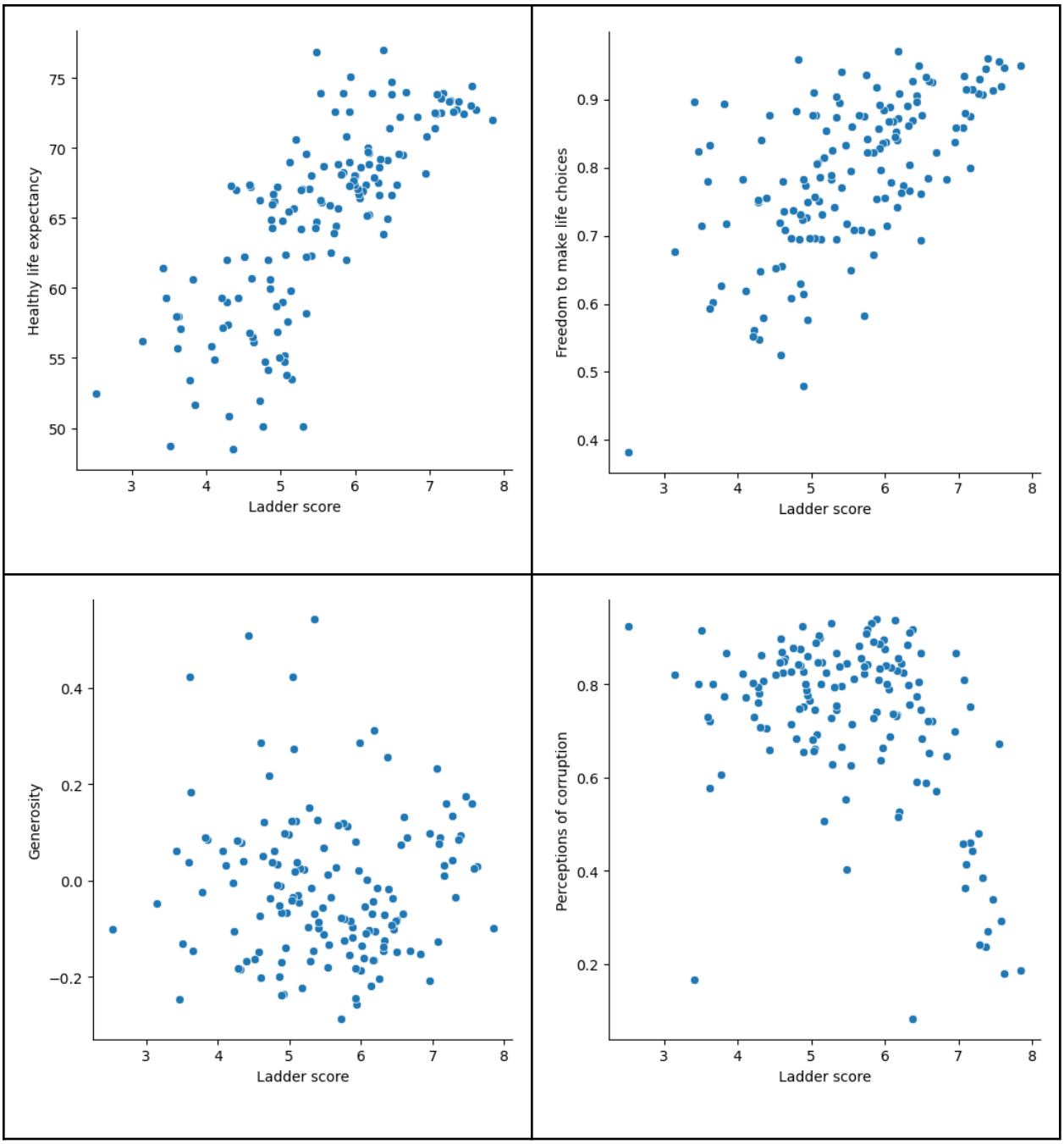
Data Set

The specific data set this paper will be using is the World Happiness Report from 2021. The World Happiness Report "reflects a worldwide demand for more attention to happiness and well-being as criteria for government policy. It reviews the state of happiness in the world today and shows how the science of happiness explains personal and national variations in happiness" (Helliwell). This set of data ranks each country using 6 factors: GDP per Capita, Social Support, Healthy Life Expectancy, Freedom to Make Life Choices, Generosity, and Perceptions of Corruption.

Data Analysis (Graphs of Each Independent Variable)

Figure 1: Ladder Score vs Variables





These 6 scatter plot graphs take into account each feature and compares it to the ladder score to determine whether a clear relationship exists. These graphs are constructed using Seaborn, a "Python data visualisation library" (see appendix for the program written to construct graphs) (seaborn). Observing the graphs, most of the graphs are linear in nature. GDP, social support, healthy life expectancy, and the freedom to make life choices all show positive linear relationships, therefore using the linear regression

may lead to the most optimal result. However, the Ladder score vs Generosity graph has a nonlinear relationship, leading to a drop in accuracy with the model. The Ladder score vs Perceptions of corruption graph does look linear in nature at first glance but closer inspection proves that most of the data is clustered. It is arguable that the Ladder score vs Perceptions of corruption graph also has a non-linear structure as most of the data is clustered in the top-middle.

Methodology

The data can be acquired through the website Kaggle as a csv (comma separated values) file (Aché). This format allows easy manipulation and visualisation of the data set since it is relatively readable.

Since the data set is relatively small with a total of 149 countries, the normal equation is a better method to use compared to gradient descent when minimising the cost function.

Using Jupyter Notebook, a web based IDE used for data science and computing, and the scikit-learn (sklearn) machine learning library in python, both the linear regression model and the regression decision tree model can be constructed. First, a clean up of the data is required.

Table 1: Pre-Cleaned Data Set

	Country name	Regional indicator	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Ladder score in Dystopia	Explained by: Log GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices	Explained by: Generosity	Explained by: Perceptions of corruption	Dystopia + residual
0	Finland	Western Europe	7.842	0.032	7.904	7.780	10.775	0.954	72.0	0.949	-0.098	0.186	2.43	1.446	1.106	0.741	0.691	0.124	0.481	3.253
1	Denmark	Western Europe	7.620	0.035	7.687	7.552	10.933	0.954	72.7	0.946	0.030	0.179	2.43	1.502	1.108	0.763	0.686	0.208	0.485	2.868
2	Switzerland	Western Europe	7.571	0.036	7.643	7.500	11.117	0.942	74.4	0.919	0.025	0.292	2.43	1.566	1.079	0.816	0.653	0.204	0.413	2.839
3	Iceland	Western Europe	7.554	0.059	7.670	7.438	10.878	0.983	73.0	0.955	0.160	0.673	2.43	1.482	1.172	0.772	0.698	0.293	0.170	2.967
4	Netherlands	Western Europe	7.464	0.027	7.518	7.410	10.932	0.942	72.4	0.913	0.175	0.338	2.43	1.501	1.079	0.753	0.647	0.302	0.384	2.798

Currently, these are the entries of the top 5 happiest countries in the world (only for 2021). However, columns such as the country name and regional indicator are unnecessary since they are classification data labels and not continuous. If included, they will result in an error when constructing the models. Therefore, these will be removed. Furthermore, the standard error and explained by data labels are unneeded in predicting the ladder score. After cleaning up the model, below is the new dataset.

Table 2: Post-Cleaned Data Set

	Logged GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
0	10.639267	0.954330	71.900825	0.949172	-0.059482	0.195445
1	10.774001	0.955991	72.402504	0.951444	0.066202	0.168489
2	10.979933	0.942847	74.102448	0.921337	0.105911	0.303728
3	10.772559	0.974670	73.000000	0.948892	0.246944	0.711710
4	11.087804	0.952487	73.200783	0.955750	0.134533	0.263218

Using only these values as the independent variables, constructing the models will be possible.

Using the cleaned up dataset, we can use the train_test_split method, “used to split our data into train and test sets” (geeksforgeeks). This sklearn method splits the dataset into four parts: X_train, X_test, y_train, and y_test. X_train and y_train are used to train the model whereas X_test and y_test are used to test it. The X and y represent features (columns) and labels (rows) respectively.

When using the method, the percentage of the data used to train compared to the data used to test the model can be specified. The value is called test_size, ranging from 0.1 to 1, with 0.1 representing 10% of the data used to test and 1 representing 100%of the data used to test the model. To maximise the accuracy of the model while still having enough data to test the results, the models will be constructed with test_size values of 0.1, 0.2, 0.3, 0.4, and 0.5.

Furthermore, the data the method chooses from the entire set will be different every time, so 5 trials of each test_size will be conducted, and then an average will be calculated.

Assessing Performance of Model

To test the accuracy of this model, the score() method can be used. This uses the R-squared, or coefficient of determination formula. This function is modelled as

$$R^2 = 1 - \frac{RSS}{TSS}$$

where R^2 is the coefficient of determination, RSS is the sum of squares of residuals, and TSS is the total sum of squares. The sum of squared residuals is the difference between predicted values of the dependent variable and observed values of the dependent variable squared. The total sum of squares is “the sum of the distance the data is away from the mean all squared” (Newcastle). The closer the R^2 is to 1, the more accurate the model is.

The error column is calculated using procedural uncertainty, taking the maximum and minimum R^2 values and dividing by 2.

Procedure

1. Import the 2021 World Happiness Report into Jupyter Notebook, and clean up the data set as shown above.
2. Use the LinearRegression and train_test_split method from the sklearn library (an installation of sklearn may be required).
    a. Use the test_size parameter, adjusting it for each manipulation as mentioned above. Conduct each manipulation for 5 trials.
3. Use the DecisionTreeRegressor and train_test_split method (use 3 for max_depth, explained later)
    a. Use the test_size parameter, adjusting it for each manipulation as mentioned above. Conduct each manipulation for 5 trials.

Using a Linear Regression Model

Since the linear model has six features, the equation for the linear regression model will be:

$$Y = \theta_0 + (\theta_1 X_1 + \theta_2 X_2 + \ldots + \theta_6 X_6) + \epsilon$$

where β_0 represents the y-intercept, $\theta_1 \sim \theta_6$ represents the slope for each of the features $X_1 \sim X_6$, and ϵ represents the random error mentioned previously.

Table 3: test_size vs R^2 Value for Linear Regression

	R^2 Value						
test_size	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average	error
0.1	0.640	0.766	0.656	0.570	0.763	0.679	0.098
0.2	0.502	0.775	0.660	0.642	0.780	0.672	0.139
0.3	0.725	0.793	0.733	0.780	0.772	0.761	0.034
0.4	0.714	0.609	0.765	0.683	0.742	0.703	0.078
0.5	0.694	0.711	0.740	0.633	0.735	0.703	0.054

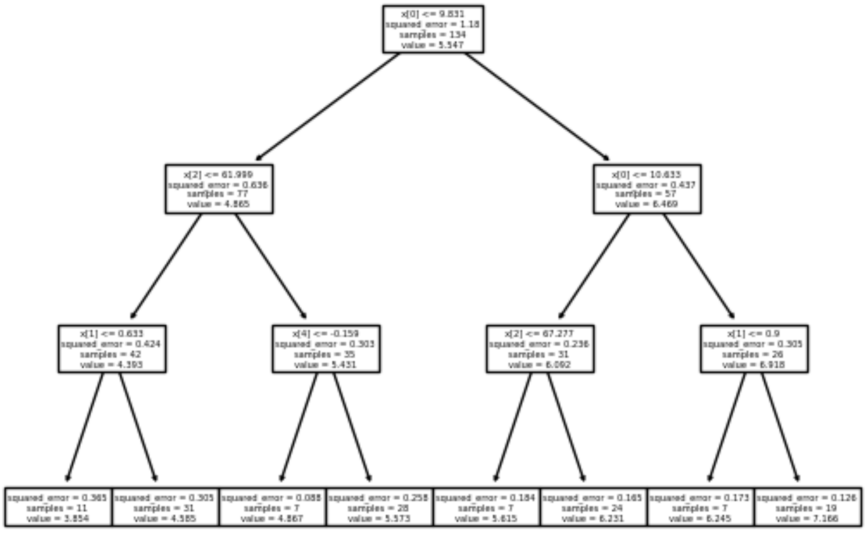
Using a Decision Tree Model

Using the train_test_split method from before along with the DecisionTreeRegressor() method, a regression tree model can be constructed. In the case of the sklearn method, if a max_depth value is not specified, it will stop only when all leaves are pure.

Table 4: test_size vs R^2 Value for Decision Trees

	R^2 Value						
test_size	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average	error
0.1	0.487	0.633	0.346	0.439	0.759	0.533	0.207
0.2	0.455	0.620	0.634	0.551	0.712	0.594	0.129
0.3	0.532	0.614	0.762	0.748	0.707	0.673	0.115
0.4	0.739	0.590	0.553	0.565	0.622	0.614	0.093
0.5	0.622	0.530	0.731	0.676	0.530	0.618	0.101

Figure 2: Visual Representation of a Decision Tree



Above is the regression tree graph for one of the trials. As mentioned before, the model splits the data set using the parameter at the top, and continues until the error is minimised. Since the max_depth was specified to 3, the tree stopped at the 3rd split. It can be seen that the squared error gradually decreases from the top branch, with the leaf nodes having the lowest error.

Analysis

Regarding the linear regression model, the highest R^2 value achieved was 0.761 for the test size of 30%, whereas the least was 0.679 for the test size of 10%, showing that it is not accurate enough to be considered reliable but does show a general trend. Furthermore, the error was the least significant for the test size of 30%, validating the result.

The reason for the inaccuracy may result in underfitting. Underfitting “occurs when the model is too simple” due to a multitude of factors (IBM). One reason is due to the simplicity of the linear regression model, which cannot accurately predict the non-linear pattern of some of the features within the data set. Another reason is the need for more inputs/entries within the data set. The World Happiness Index, as mentioned before, only contains 149 rows for 2021, making it a relatively small data set.

In contrast, the decision tree model was less accurate compared to the linear regression model on average. The highest R^2 value achieved was 0.673 for the test size of 30%, the exact same as the linear regression model. This shows that the 30% sample test size is the best ratio for this data set in terms of training to testing is 7:3. The least accurate test_size was 10%, also the same as the linear regression model.

Furthermore, on average, the procedural uncertainty (error) was large for decision trees compared to the linear regression model, ranging all the way from 0.093 up to 0.207. A procedural uncertainty of 0.207 translates to about 38.8% error, a considerably large error.

The possible explanation to this lack of predictability is the opposite from the explanation for the unreliability of the linear regression model: overfitting. Decision trees are "prone to overfitting" (Patani), a phenomenon where "the model cannot generalise and fits too closely to the training dataset," causing a low accuracy when testing it (Amazon). There are a few reasons to explain why overfitting is happening to this specific data set. First is that the model fits too closely as the training data size is small, similar to the reason for underfitting. Another reason for this phenomenon is that it contains too many outliers or irrelevant information. From the graphs before, we can see that there are quite a few points on the graphs that stray away from the general trend. However, the best case scenario to this is that the model should recognize that it is an outlier using the best fit. The reason this does not happen is that the model complexity is too high. The max_depth specified (3) may have been too high that the model overfitted (Amazon).

Evaluation

Considering the best values of both models, the R^2 values of 0.761(76.1%) and 0.673(67.3%) respectively for linear regression and decision trees answers the research question of to what extent regression techniques in predictive analytics can be used to predict happiness levels of countries. The linear regression model serves to be much more applicable compared to the decision trees, but both models can be improved. There are some limitations of the methodology used.

1. First is the data itself. As discussed before, the Generosity vs Ladder score graph does not have a correlating relationship. However, not considering the data would undermine the fundamental purpose of the paper as the results would be inconclusive, so a non-linear approach to predict the values would need to be researched and implemented.
2. Another option would be to consider all of the outliers within each column and manually remove ones that do not align with the general trend to improve the accuracy of the decision tree model. However, at the same time, this undermines the purpose of using machine learning, as much human intervention would be required to erase the noise.
3. Finally, the World Happiness Report has been conducted for many years. Using previous years' data to increase the total number of rows can help increase the accuracy of the two models since the number of data points would increase. Both the training and testing data will increase as a result, improving both accuracy and precision.
4. With the decision tree model, choosing the most accurate model from the training data to testing data ratio, then proceeding to change the max_depth value would have served to increase the accuracy as well as the uncertainty of the model. Predetermining the max_depth to 3 may have been a hindrance.

However, the results seem to be promising since the error for the linear regression is quite low, proving the precision of the model. An extension would be to utilise the technology used in each model and combining them two takes the best of both worlds. The model is called a Linear Model Tree (LMT), which is a "decision tree with linear models at its nodes." Since it can split the population into subcategories then form a linear relationship, it can take into account different behaviours with certain countries/data points. Like linear regression, it "can work well with a modest amount of data," which may be able to solve the problem of overfitting and underfitting (Dillard).

It is important to note that there are some aspects to the approach of this investigation that are noteworthy. The use of both linear regression and decision trees offered different ways to answer the research question. Furthermore, the use of different test sizes served to improve both precision and

accuracy of the results rather than predetermining a test size. Finally, the selection of which model to use as well as the selection of the different loss functions served to make a model with greater accuracy.

Conclusion

The aim of the paper was to answer if predictive analytics is a valid approach to predict the happiness levels of countries given parameters, and the findings above support the claim that it is valid, given further improvements. The methodology above is a solid baseline for starting, but can be further improved with the suggestions listed above.

However, beyond the scope of this specific question, it proves that predictive analytics can be used not only for business practices but for unconventional purposes such as the one in this paper.

This is not the first investigation to do this however. A similar study was conducted with the use of the Better Life Index, and the accuracy of their model (random forest) came out to 90%, proving that with tuning reliability is possible (Kaur).

An interesting extension to this paper would be to focus on a specific demographic instead of multiple countries, an example being a school. Data of students' and teachers' happiness can be collected, as well as other parameters such as weather, number of assignments/tests on that day, the time of day a person woke up, etc… Predictive analytics can be used to investigate whether these day-to-day variables can affect the happiness of one. This research relates to fields of psychology, showing that predictive analytics can be used in applications of other fields.

Overall, this investigation serves as an insight to the various real-life applications of predictive analytics.

Works Cited

Achè Mathurin. “World Happiness Report up to 2022.” Kaggle, 19 Mar. 2022, www.kaggle.com/datasets/mathurinache/world-happiness-report?select=2022.csv.

Amazon. “What Is Overfitting?”, aws.amazon.com/what-is/overfitting/#:~:text=Overfitting%20is%20an%20undesirable%20machine,on%20a%20known%20data%20set.

Carnegie Mellon. “Lecture 10: Regression Trees - Carnegie Mellon University.” Statistics & Data Science Dietrich College of Humanities and Social Sciences, 2006, www.stat.cmu.edu/~cshalizi/350-2006/lecture-10.pdf.

Dan. “Linear Regression vs Decision Trees.” |, 1 Sept. 2020, mlcorner.com/linear-regression-vs-decision-trees/.

Dillard, Logan. “The Best of Both Worlds: Linear Model Trees.” Medium, Convoy Tech, 14 Mar. 2018, medium.com/convoy-tech/the-best-of-both-worlds-linear-model-trees-7c9ce139767d#:~:text=Linear%20model%20trees%20combine%20linear,insights%20than%20either%20model%20alone.

Foley, Michael “My Data Science Notes.” Chapter 8 Decision Trees, 26 July 2020, bookdown.org/mpfoley1973/data-sci/decision-trees.html.

GeeksforGeeks “How to Split the Dataset with Scikit-Learn’s Train_test_split() Function.” GeeksforGeeks, 29 June 2022, www.geeksforgeeks.org/how-to-split-the-dataset-with-scikit-learns-train_test_split-function/.

GeeksforGeeks, “ML: Normal Equation in Linear Regression.” GeeksforGeeks, GeeksforGeeks, 10 June 2023, www.geeksforgeeks.org/ml-normal-equation-in-linear-regression/.

Halton, Clay. “Predictive Analytics: Definition, Model Types, and Uses.” Investopedia, Investopedia, 6 Apr. 2023, https://www.investopedia.com/terms/p/predictive-analytics.asp.

Helliwell, John. “About.” World Happiness Report, 2023, worldhappiness.report/about/.

IBM “What Is a Decision Tree”, www.ibm.com/topics/decision-trees#:~:text=data%20mining%20solutions-,Decision%20Trees,internal%20nodes%20and%20leaf%20nodes. Accessed 23 Sep. 2023.

IBM, “What Is Machine Learning?”, www.ibm.com/topics/machine-learning. Accessed 23 Sep. 2023.

IBM, “What Is Predictive Analytics?” https://www.ibm.com/topics/predictive-analytics#:~:text=Predictive%20analytics%20is%20a%20branch,to%20identify%20risks%20and%20opportunities.

IBM. “What Is Underfitting?” IBM, www.ibm.com/topics/underfitting#:~:text=the%20next%20step-,What%20is%20underfitting%3F,training%20set%20and%20unseen%20data. Accessed 23 Sep. 2023.

Khan. “Gradient Descent (Article).” Khan Academy, www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/optimizing-multivariable-functions/a/what-is-gradient-descent. Accessed 23 Sept. 2023.

Kaur, Maninder, et al. “Supervised Machine-Learning Predictive Analytics for National Quality of Life Scoring.” Applied Sciences, vol. 9, no. 8, Apr. 2019, p. 1613. Crossref, https://doi.org/10.3390/app9081613.

Kumar, Vaibhav, and M. L. Garg. "Predictive analytics: a review of trends and techniques." International Journal of Computer Applications 182.1 (2018): 31-37.

Luo, Shuyu. “Optimization: Loss Function under the Hood (Part I).” Medium, Towards Data Science, 14 Oct. 2018, towardsdatascience.com/optimization-of-supervised-learning-loss-function-under-the-hood-df1791391c82.

Patani, Marmik. “Decision Tree - for Beginners.” Medium, Analytics Vidhya, 27 Jan. 2021, medium.com/analytics-vidhya/decision-tree-for-beginners-dd966beb0a8b#:~:text=A%20decision%20tree%20is%20a%20graphical%20representation%20of%20all%20possible,decision%20based%20on%20certain%20conditions&text=Decision%20Trees%20are%20versatile%20machine,widely%20used%20to%20supervised%20learning.

Schneider, Astrid et al. “Linear regression analysis: part 14 of a series on evaluation of scientific publications.” Deutsches Arzteblatt international vol. 107,44 (2010): 776-82. doi:10.3238/arztebl.2010.0776

Scikit Learn. “1.10. Decision Trees.” Scikit, scikit-learn.org/stable/modules/tree.html#:~:text=Decision%20Trees%20(DTs)%20are%20a,as%20a%20piecewise%20constant%20approximation. Accessed 23 Sep. 2023.

Seaborn “Statistical Data Visualization#”, seaborn.pydata.org/. Accessed 23 Sept. 2023.

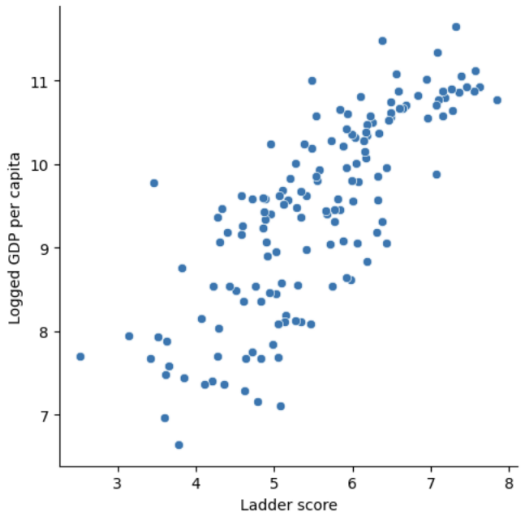
Shah, Saily. “Cost Function Is No Rocket Science!” Analytics Vidhya, 18 Aug. 2023, www.analyticsvidhya.com/blog/2021/02/cost-function-is-no-rocket-science/.

University, Newcastle. “Coefficient of Determination, R-Squared.” Numeracy, Maths and Statistics - Academic Skills Kit, www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html#:~:text=%C2%AFy. Accessed 23 Sep. 2023.

Appendix

Example of Code for Graph Generation

```
In [6]: sns.relplot(x='Ladder score',y='Logged GDP per capita', data=data_2021)
Out[6]: <seaborn.axisgrid.FacetGrid at 0x14a7fa770>
```



Code used to Generate Linear Regression Model

```
In [12]: from sklearn.linear_model import LinearRegression
         from sklearn.model_selection import train_test_split
         from sklearn.tree import DecisionTreeRegressor

In [13]: train = data_2021.drop(['Ladder score', 'Country name', 'Regional indicator', 'Standard error of ladder score', 'upperw
         test = data_2021['Ladder score']

In [483]: X_train, X_test, y_train, y_test = train_test_split(train, test, test_size = 0.5)

In [484]: regr = LinearRegression()

In [485]: regr.fit(X_train, y_train)

Out[485]: LinearRegression()
          In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
          On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [486]: pred = regr.predict(X_test)

In [487]: pred

Out[487]: array([4.1870584 , 5.4040718 , 3.85891856, 6.09496663, 5.05490786,
                 6.32375963, 5.49595911, 6.52255092, 6.06826669, 5.25334721,
                 5.74583687, 5.92185818, 5.33558127, 4.09519259, 4.88185835,
                 4.27483777, 5.77306158, 6.90134076, 5.68467477, 5.70493987,
                 5.73613697, 5.54693461, 6.04428   , 4.5662626 , 4.81103889,
                 7.00484729, 6.21299823, 5.5660034 , 5.81514081, 4.12635199,
                 6.30472067, 4.90621466, 5.61966975, 3.81715464, 4.50111226,
                 4.38636918, 4.35473578, 7.14295773, 2.59277794, 4.97387609,
                 6.43367076, 5.63523493, 6.22822784, 3.46265263, 4.36664972,
                 7.27004973, 6.04477633, 5.90805819, 5.64347029, 5.72532517,
                 7.04134406, 5.64111783, 5.08647488, 5.51202048, 6.14464762,
                 4.67603785, 6.89707107, 3.81397613, 4.19493899, 3.60543046,
                 5.42647995, 4.43201835, 5.81856646, 4.72904421, 5.39478158,
                 5.42661386, 5.67795966, 5.19625788, 5.76789645, 5.42418471,
                 5.08776959, 5.5183971 , 3.95974175, 5.43200116, 4.78195135])

In [488]: regr.score(X_test, y_test)

Out[488]: 0.7410946083130763
```

Code used to Generate Regression Decision Tree Model

In [489]:
```
TreeRegr = DecisionTreeRegressor(max_depth = 3)
```

In [490]:
```
TreeRegr.fit(X_train, y_train)
```

Out[490]:
```
DecisionTreeRegressor(max_depth=3)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [491]:
```
TreeRegr.predict(X_train)
```

Out[491]:
```
array([4.5303125 , 5.44489474, 6.1585    , 7.15038462, 5.44489474,
       7.15038462, 5.44489474, 4.5303125 , 5.44489474, 6.09683333,
       4.5303125 , 7.15038462, 5.44489474, 4.5303125 , 7.15038462,
       6.09683333, 4.5303125 , 3.662     , 5.44489474, 7.15038462,
       5.44489474, 7.15038462, 4.7446    , 3.662     , 5.44489474,
       5.44489474, 3.662     , 6.09683333, 5.481     , 5.44489474,
       4.5303125 , 5.481     , 6.09683333, 4.7446    , 6.09683333,
       6.1585    , 7.15038462, 7.15038462, 6.09683333, 4.5303125 ,
       6.09683333, 4.5303125 , 7.15038462, 5.44489474, 4.5303125 ,
       6.09683333, 4.5303125 , 7.15038462, 6.09683333, 4.7446    ,
       5.44489474, 5.44489474, 4.5303125 , 6.09683333, 4.7446    ,
       5.44489474, 5.44489474, 7.15038462, 3.662     , 5.44489474,
       6.09683333, 5.481     , 5.44489474, 4.7446    , 7.15038462,
       4.5303125 , 5.44489474, 7.15038462, 4.5303125 , 4.5303125 ,
       5.44489474, 4.5303125 , 4.5303125 , 6.09683333])
```

In [492]:
```
TreeRegr.score(X_test, y_test)
```

Out[492]:
```
0.5297540253765038
```

In [235]:
```
from sklearn import tree

XYZ = tree.plot_tree(TreeRegr)
```