Understanding Global Happiness Patterns: Machine Learning Perspectives on the World Happiness Report

Arshia Sharifi [redacted]

Ben Costas [redacted]

CPS844: Data Mining

Prof. Elodie Lugez

March 4, 2024

Classification Methods:

1. Decision Tree - **Ben**
2. Artificial Neural Network - **Ben**
3. Naive Bayesian (Gaussian) - **Arshia**
4. K-Nearest Neighbor - **Arshia**
5. Support Vector Machines (SVD) (**Arshia, Ben**)

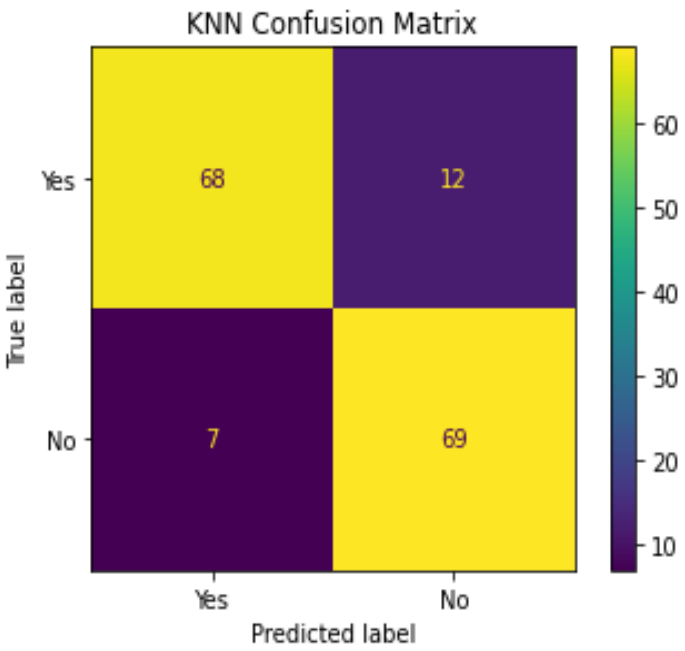
Introduction: The World Happiness Report, sourced from Kaggle [1], captures happiness scores and contributing factors across 156 countries. Moreover, in our analysis, we have chosen to focus on a subset of attributes from the original dataset, excluding 'Overall Rank', 'Country or Region', and 'Perception of Corruption'. This deliberate selection aims to streamline our analysis and avoid subjective political assessments. Additionally, we have discretized the continuous 'Score' attribute using a benchmark of 5.43, which is aligned with the median across all of the instances. The primary objective is to explore how socioeconomic factors such as GDP per Capita, social support, freedom to make life choices, and generosity influence overall happiness scores. Through predictive modeling and data-driven analysis, we seek to uncover the most effective classification method for identifying the most important features in determining happiness.

K-Nearest Neighbor: K-Nearest Neighbors (KNN) is a simple and intuitive classification algorithm that works by assigning a class label to a new data point based on the majority of its k nearest neighbors in the feature space. We trained our model on the 2019 World Happiness Report and then tested it on the 2018 report due to the limited size of the dataset, which contains only 156 records. To determine the optimal value of k, we performed cross-validation with k values ranging from 1 to 31, recording the mean accuracy for each k value. After conducting cross-validation, we found that the best-performing k value was 10, which resulted in an accuracy of 88%. The confusion matrix (Figure 1) illustrates the classification performance of our KNN model, with values [68, 12] and [7, 69] indicating the counts of true positives, false positives, false negatives, and true negatives, respectively.

Determining Most Important Features: We employed a permutation-based method to identify the most important features. By permuting each feature one at a time and measuring the resulting change in accuracy, we assessed the impact of individual features on the model's performance. For instance, in the iteration assessing the 'GDP per capita' feature, we shuffled its values while keeping others constant and evaluated the model's accuracy. The difference between the original accuracy and the accuracy obtained with the permuted feature reflected the 'GDP per capita' influence. This process was repeated for each feature, determining the two features causing the largest accuracy decrease upon permutation. In our analysis, 'GDP per capita' and 'social

support' emerged as the most influential features, significantly affecting predictive accuracy. On the contrary, 'generosity' and 'freedom to make life choices' showed minimal impact, indicating their lesser relevance in predicting happiness levels.

Lastly, Figure 2 depicts a heatmap showcasing the distribution of data points based on GDP per capita and social support. This visualization provides insights into the clustering patterns of countries concerning these two critical factors contributing to happiness levels. The KNN decision boundary overlaid on the heatmap demonstrates how the model distinguishes between different happiness levels based on mapping GDP per capita and social support. The yellow region represents “Yes” and vice versa for the purple.



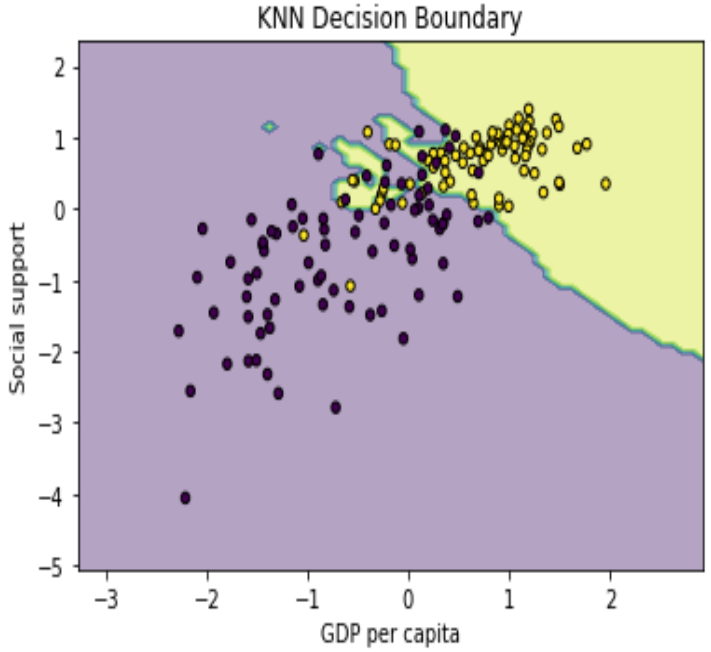


Figure 1. KNN Confusion Matrix

Figure 2. KNN Heatmap of the Most Important Features

Naive Bayes: Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. We utilized the Gaussian Naive Bayes method for our analysis, considering the discretized nature of the 'Score' feature into binary categories of 'Yes' or 'No'. Following the same testing and training methodology as KNN, our model

achieved an accuracy of 85%, with the confusion matrix [65, 15] and [8, 68] indicating true positives, false positives, false negatives, and true negatives, respectively (Figure 3).

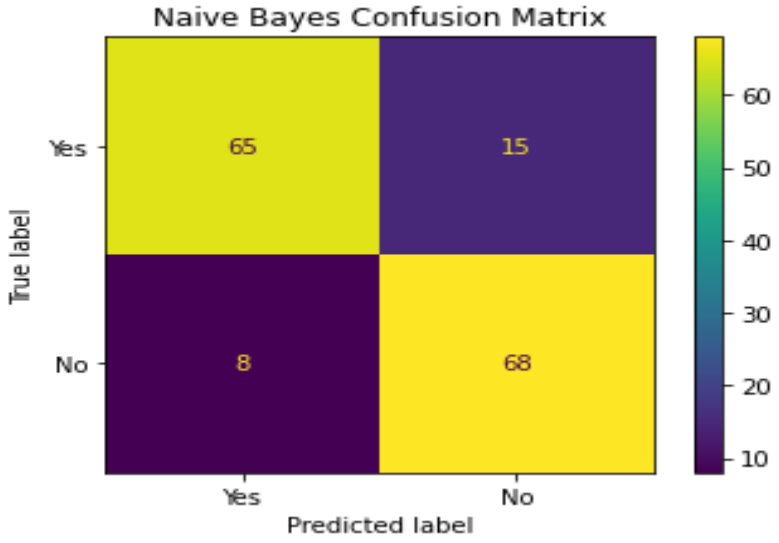
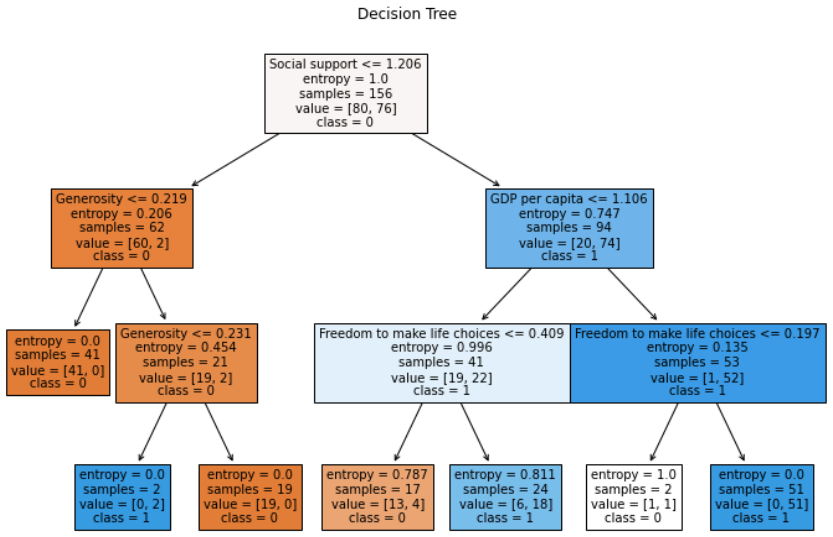


Figure 3. Naive Bayes Confusion Matrix

Determining Most Important Features: In our feature importance analysis using permutation importance, we randomized the values of each feature one at a time and observed the resulting change in accuracy. By calculating the mean importance from these permutations, we gained insights into the significance of each feature. It's important to note that the list [0.129, 0.126, 0.049, 0.009] represents the mean importance, indicating the average impact of each feature on the model's performance. While these specific values may vary slightly across different runs, they consistently reflect the relative importance of the features in predicting happiness levels. In our analysis, ‘GDP per capita’ and ‘social support’ emerged as the two most influential features.

Decision Tree: The Decision Tree is a simple classification algorithm. It employs a flowchart-like tree structure to determine an output based on a set of inputs. Within the structure, each internal node represents a decision based on a feature, while each branch represents the outcome of that decision. The depth of the tree is variable and can depend on the complexity of the data being inputted. In our dataset, we are only testing the classifier on 4 numerical weighted attributes to determine a binary output of ‘Yes’ or ‘No’ for our ‘Score’ feature. Thus, a

maximum depth of 3 is ideal to reduce any bias, and errors, and prevent cases of overfitting. We used the entropy criterion to determine the best measure of node impurity as we want to interpret the average amount of information needed to determine whether the score is 'Yes' or 'No'. After training and testing our model, we achieved an accuracy of 85%. Our confusion matrix depicted 64 true positives, 16 false positives, 8 false negatives, and 68 true negatives.



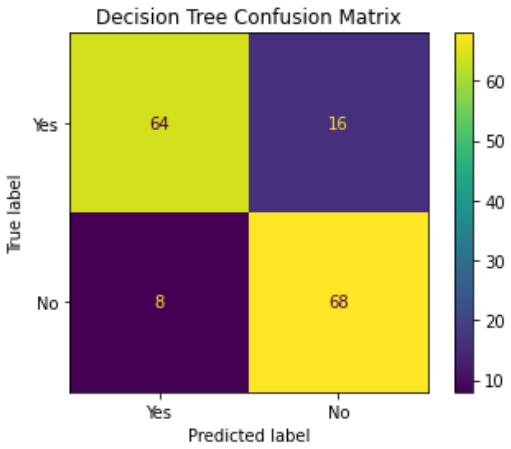


Figure 4: Decision Tree

Figure 5: Decision Tree Confusion Matrix

Determining Most Important Features: The feature importance displayed below depicts 'GDP per capita' to have an importance of 0.183, 'Social support' has an importance of 0.602, 'Freedom to make life choices' has an importance of 0.108, and 'Generosity' has an importance of 0.105. This was calculated based on the information gain brought about by each feature, otherwise known as entropy importance. Based on the conclusions above, for this particular classification model, 'Social support' is depicted as the most important feature whereas the remaining features aren't as significant for predicting the overall happiness score.

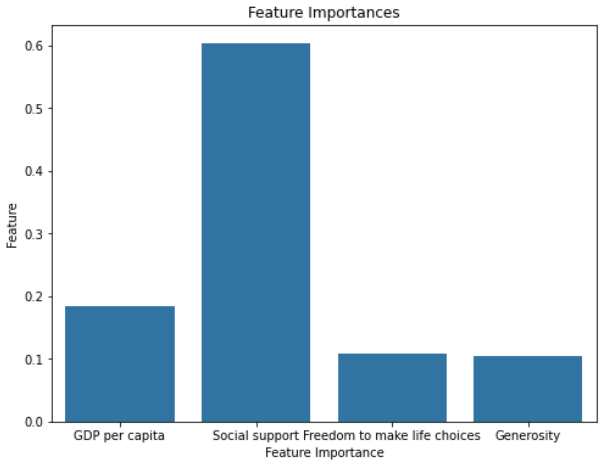


Figure 6: Decision Tree Feature Importance

Artificial Neural Network: Artificial neural networks are an intuitive and complex classification model. Its structure mimics the functions that occur when our brain processes information. ANNs consist of interconnected layers of nodes (which represent neurons), where each node performs a weighted sum of its inputs, applies an activation function to the result, and passes the output to the nodes in the following layer. Our activation function uses the logistic sigmoid function, as the minimum and maximum output values of this function are 0 and 1, which corresponds to the binary values we want to predict in our happiness score. In our dataset, we are only inputting 4 numerical categories into our ANN and producing a binary result. It is important to not overly complicate our ANN as this could affect the computation. Thus, we used 5 hidden layers and a maximum of 400 iterations to compute our classification model. Our ANN achieved an accuracy of 86%. Our confusion matrix depicted 69 true positives, 11 false positives, 11 false negatives, and 65 true negatives.

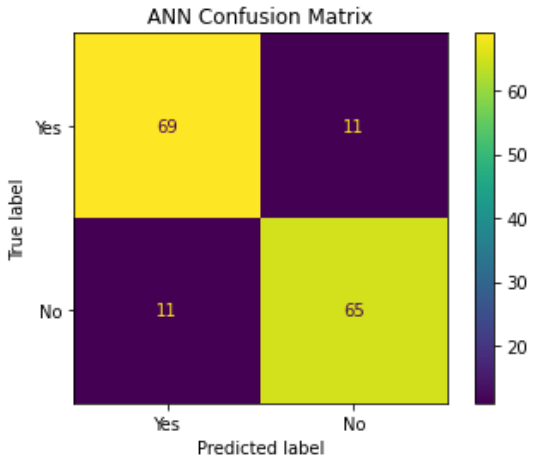


Figure 7: ANN Confusion Matrix

Determining Most Important Features: To determine the most important features, we extracted the weights connecting inputs to the first hidden layer and then calculated the feature importances as the sum of absolute weights for each feature. We then normalized the feature importance and plotted it in the graph below. From what we can observe, 'GDP per capita' and 'Social support' were the most prominent features, with 'Freedom to make life choices' behind by roughly 0.05 and 'Generosity' at 0.15. It should be noted that there is more balance to this feature importance chart than for the decision tree. This demonstrates that the ANN complexity can compensate for all input categories and base its final output on an even distribution.

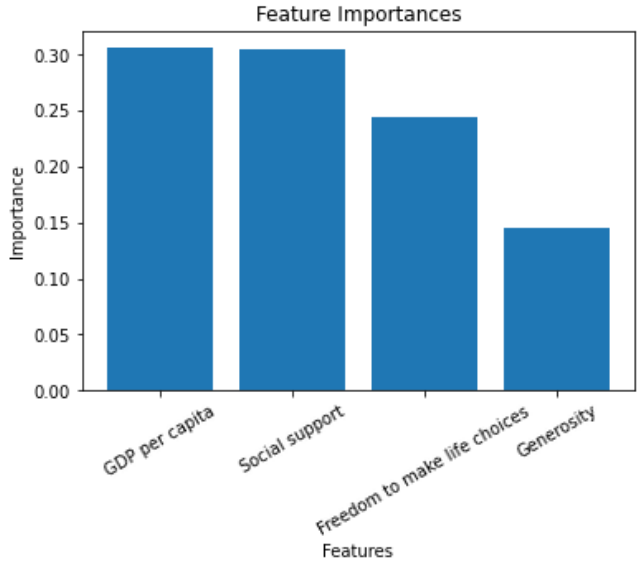


Figure 8: ANN Feature Importances

Support Vector Machines: Support Vector Machine is a classification algorithm that operates by finding the optimal hyperplane to separate data points into different classes in the feature space. It achieves this by maximizing the margin between the hyperplanes and the closest data points. When a clear linear hyperplane can't be determined, SVM utilizes kernel functions such as the Gaussian Radial Basis Function (RBF) kernel to transform the input data into higher dimensional spaces. In our analysis, we converted the class attributes "Yes" and "No" to binary labels 1 and 0 respectively, and achieved an accuracy of 87%. The resulting confusion matrix, with [68, 12] and [8, 68] manifests the model's performance in terms of true positives, false positives, false negatives, and true negatives respectively.

Determining Most Important Features: To determine the most important features of the SVM classification model, we extracted the support vectors which are the data points closest to the margin. Each of these support vectors contained a dual coefficient, which signifies its importance in determining where the margin would be located. Afterwards, we calculated the absolute dot product of the dual coefficients and support vectors which provided us with a percentage representing the precedence of each feature. From the graph below, it is depicted that 'GDP per capita' and 'Social support' were the most important features influencing the outcomes produced by the SVM classification model.

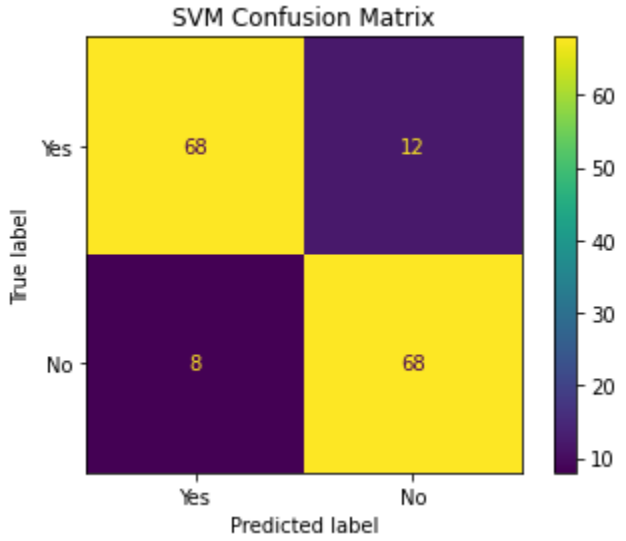


Figure 9. SVM Confusion Matrix

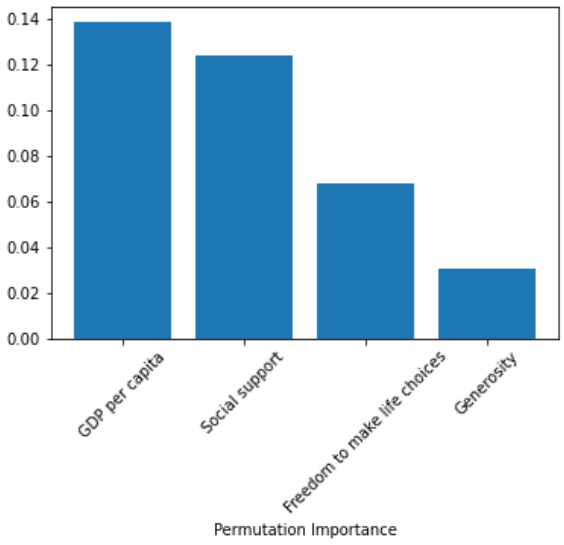


Figure 10. SVM Feature Importance

Which Model Does The Best Job:

After analyzing and comparing the results produced from all 5 classification models, we determined that the model that does the best job at predicting the results is KNN. The better performance can be credited to the fact that KNN is most suitable for smaller datasets like ours, consisting of 156 records, KNN does not require assumptions about the data distribution, is computationally efficient for such scale, and achieved the highest accuracy with 88%. When comparing the performance of the other classification models to KNN, the decision tree has an accuracy of 85% but its prediction is heavily influenced by the 'Social support' category. SVM has an accuracy of 87% but its output is influenced by 'GDP per capita' and 'Social support'. NB is similar to SVM as it has an 85% accuracy with higher precedence for 'GDP per capita' and 'Social support' compared to the other categories. ANN has a lower accuracy than KNN with 86%, however, each of the categories influences the outcome. Likewise, the ANN can be optimized further by adding more layers which can improve the accuracy of the predictions, but this comes with the price of higher computing and calculation costs.

References

1. Abigail Larion, World Happiness Report, https://www.kaggle.com/datasets/unsdsn/world-happiness
2. Jason Brownlee, How to Calculate Feature Importance With Python, https://machinelearningmastery.com/calculate-feature-importance-with-python/