Report on

Predicting life expectancy: Performance of various machine learning algorithms

Submitted in partial fulfillment of the requirements of the course

Machine Learning for Data Science

Course Code: WM-ASDS22

Batch # 8, Section # B

Professional Master in Applied Statistics and Data Science under Weekend Program (WP-MASDS)

Department of Statistics

Jahangirnagar University

Predicting life expectancy: Performance of various machine learning algorithms

List of Group Members

Name	ID Number
Abul Hasan Muhammad Shahadat Ullah	201900101007
Chandan Kumar Roy	20228039
Razib Mustafiz	20228062
Akram Hossain	20228071
S. M. Yusuf	20228058

Machine Learning for Data Science

Course Code: WM-ASDS22

Batch # 8, Section # B

Predicting life expectancy: Performance of various machine learning algorithms

Abstract

This paper presents the train and test scores of various machine learning models on a life expectancy dataset. The study aims to understand the determinants of life expectancy by identifying key predictors using machine learning algorithms. The models were trained using data from the World Health Organization (WHO) and the World Bank, and the results show that spending, education, income, HIV etc. are significant determinants of life expectancy. The SVR model performed the best in terms of both train and test scores, followed by the XGB model, while the decision tree model did not perform well. The findings have important implications for policymakers and public health practitioners to develop interventions to improve population health and well-being. However, it is crucial to consider other factors such as model complexity, interpretability, and computational efficiency when choosing a suitable model.

Table of Contents

List of Tables and Figures

List of Figures

Predicting life expectancy: Performance of various machine learning algorithms

1 Introduction:

Life expectancy is an important measure of population health and can provide valuable insights into the health status of populations. With the increasing availability of healthcare data and advances in machine learning techniques, predicting life expectancy through machine learning has become an active area of research. Machine learning algorithms can leverage large and complex datasets to identify patterns and relationships between different features and outcomes. By analyzing health and demographic data, machine learning models can identify risk factors and predict life expectancy with high accuracy. Such models have the potential to improve health outcomes by providing personalized recommendations for individuals and informing public health policies.

Previous studies have shown promising results for using machine learning algorithms for predicting life expectancy. Existing studies used machine learning algorithms to predict life expectancy based on health and demographic data from multiple countries and found that gradient boosting outperformed other algorithms in terms of predictive accuracy. However, it is important to evaluate the performance of these algorithms in different contexts and datasets to determine their generalizability and potential for practical use.

In this study, we aim to compare the performance of various machine learning algorithms for predicting life expectancy. We will use data from the World Health Organization and the World Bank to develop predictive models using different algorithms, including random forest, support vector regression, and gradient boosting. We will evaluate the performance of these algorithms using metrics such as mean absolute error, root mean square error, and R-squared. By comparing the performance of these algorithms, we can identify the best-performing algorithm for predicting life expectancy and provide insights into the factors that influence life expectancy.

1.1 Objective of the study:

The objective of this study is to explore and compare the performance of various machine learning algorithms in predicting life expectancy.

2 Literature Review:

Life expectancy prediction is a crucial area of research as it can assist in developing health policies, planning health interventions and medical advancements. Machine learning (ML) algorithms have been extensively employed for predicting life expectancy as they can learn complex relationships between the predictors and the outcome.

Traditional methods of predicting life expectancy, such as demographic and medical factors, have shown limited accuracy (Suh et al., 2021). Machine learning algorithms have emerged as a promising approach for predicting life expectancy, as they can analyze large and complex datasets to identify patterns and relationships between different features and outcomes (Wang et al., 2020).

Several studies have used machine learning algorithms to predict life expectancy based on various health and demographic factors, such as age, sex, smoking status, and medical history. For example, a study by Li et al. (2021) used a random forest algorithm to predict life expectancy based on demographic and clinical data. The study found that the random forest model achieved high accuracy in predicting life expectancy. Other studies have explored the use of deep learning algorithms, such as neural networks, for predicting life expectancy. For instance, a study by Kim et al. (2020) used a convolutional neural network to predict life expectancy based on medical images. The study found that the neural network model outperformed traditional methods in predicting life expectancy. A study conducted by Kumar et al. (2021) used four ML algorithms, namely the DT, RF, SVM and artificial neural network (ANN), to predict life expectancy in India. The study used demographic, socioeconomic and health-related variables to predict life expectancy. The study found that the SVM algorithm outperformed the other three algorithms with an accuracy of 0.86. Study conducted by Bhargava et al. (2020), three ML algorithms, namely the linear regression (LR), SVM and RF, were used to predict life expectancy in India. The study used demographic and socioeconomic variables to predict life expectancy. The study found that the LR algorithm outperformed the other two algorithms with an accuracy of 0.78.

Despite the promising results of these studies, there are still several challenges to using machine learning algorithms for life expectancy prediction, such as data availability, data quality, and ethical considerations. Further research is needed to address these challenges and explore the potential of machine learning in predicting life expectancy.

3 Data and methodology

3.1 Data source:

Data on life expectancy has been collected from secondary data source data from the World Health Organization (WHO) and the World Bank. The panel dataset consist of 183 countries over the period 2005 to 2015 with highest number of observations is 2938. The summary statistics of data is as follows:

Table 1 Descriptive statistics (See Appendix A)

3.2 Explanatory data analysis

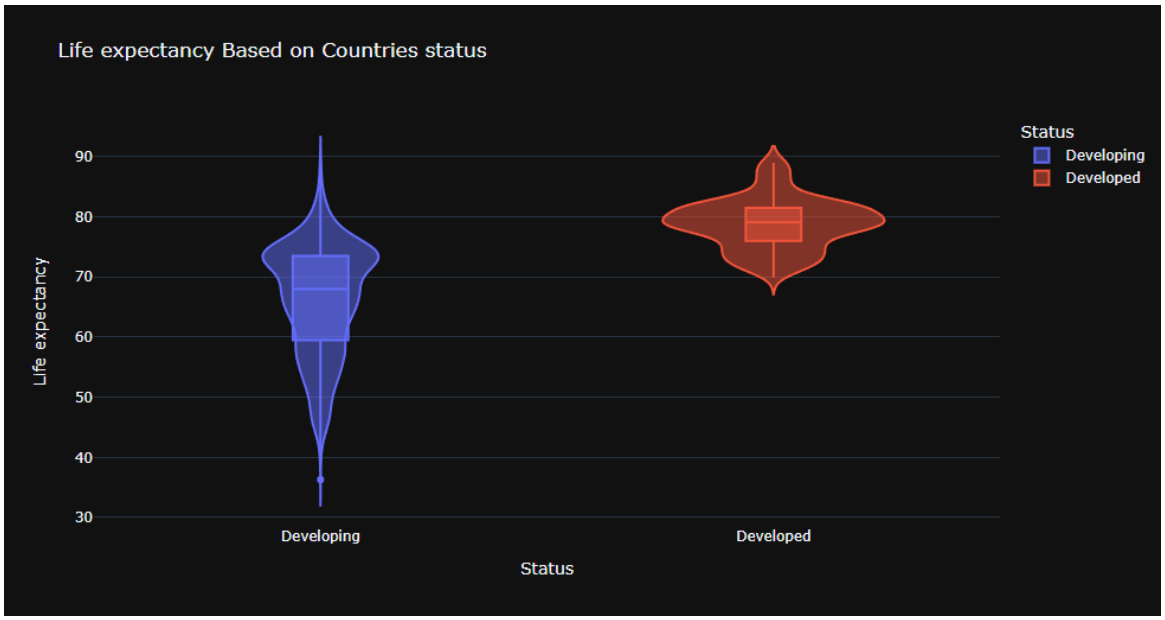
Initial Investigation and analysis of the selected dataset are done in this section. This section has multiple subsections. The dataset has 22 columns and 2938 rows. Three of the columns have nominal data while the rest has continuous value. Several graphs and visualization

techniques were used to fully comprehend the underlying relationship between the data. Here are some examples:

3.3 Violin Plot

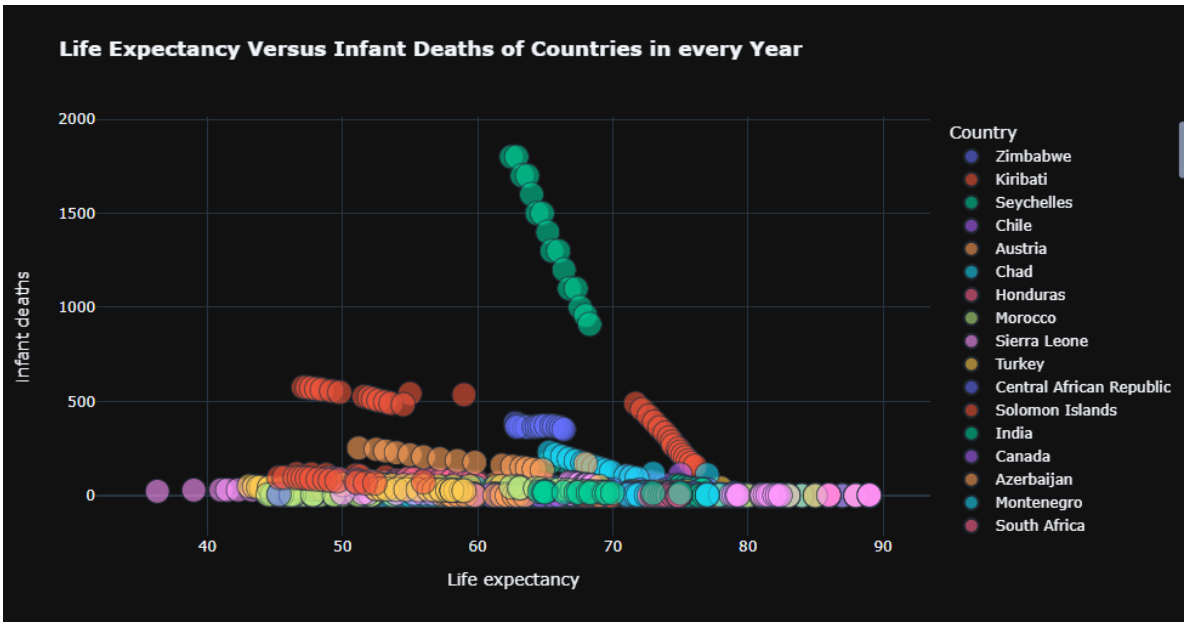
A violin plot is a type of data visualization that combines the features of a box plot and a kernel density plot to display the distribution of a continuous variable. The plot takes its name from its shape, which resembles a violin. Form this graph we can know that the life expectancy in developed countries is much higher, usually around 80. Life expectancy in developing countries, on the other hand, begins at 30 and is most dense around 73. The difference in life expectancy between developed and developing countries can be attributed to a variety of factors, including differences in healthcare, education, income, access to clean water and sanitation, and lifestyle factors such as diet and exercise. According to the World Health Organization (WHO), life expectancy in developed countries is generally higher than in developing countries (WHO, 2021). Developed countries have more access to healthcare, education, and resources that can lead to better health outcomes and longer life expectancies. In contrast, developing countries often face challenges such as poverty, limited healthcare access, and poor sanitation, which can contribute to lower life expectancies (WHO, 2021).

Figure 1 Violin Plot



4 Scatter Plot

Figure 2 Scatter Plot Diagram

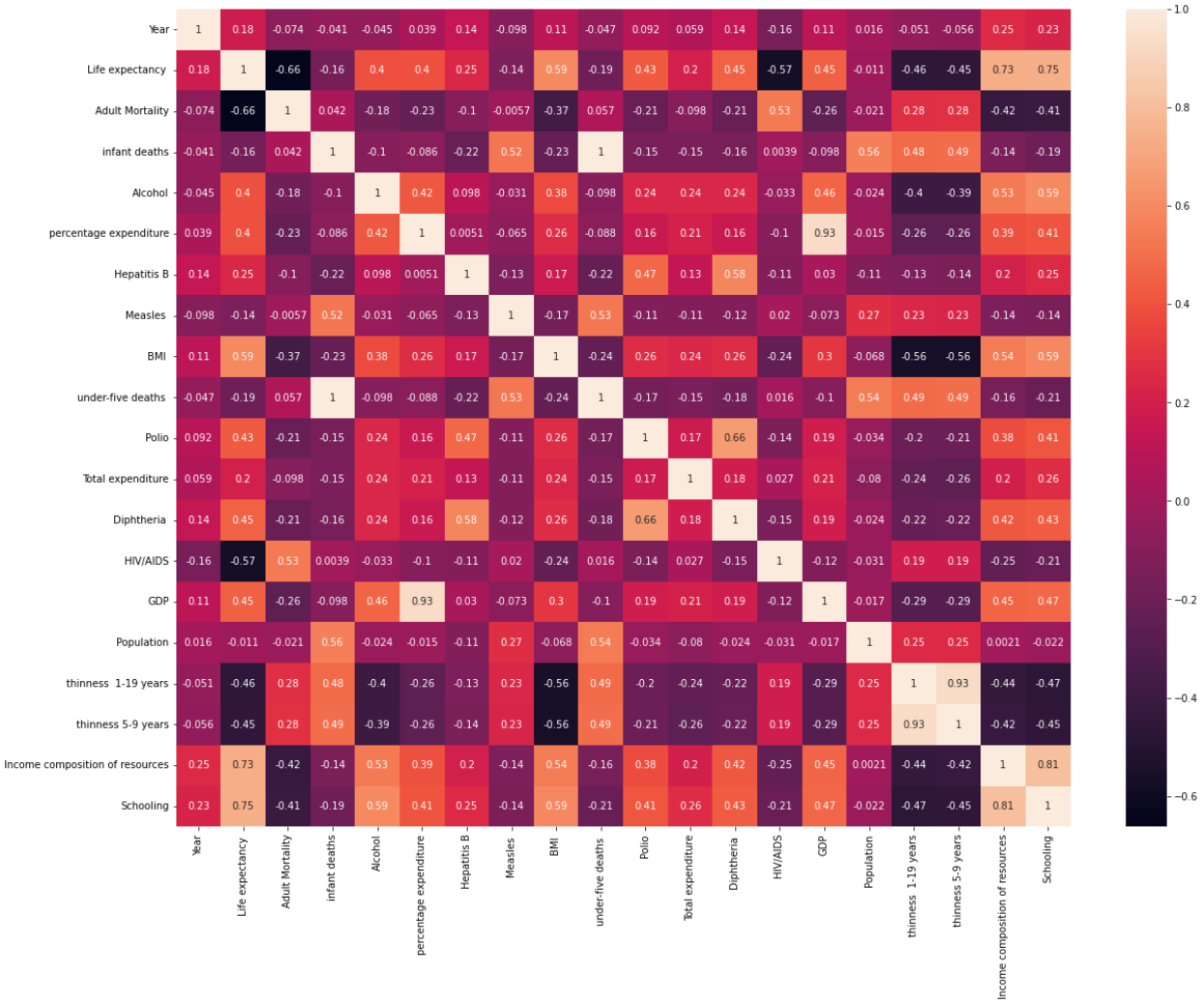


This graph depicts the relationship between infant death and life expectancy. Infant mortality appears to have a slight negative correlation, implying that a reduction in infant death will result in a slight increase in life expectancy. A slight negative correlation between infant mortality and life expectancy would imply that as infant mortality decreases, life expectancy tends to increase. This relationship makes sense because infant mortality is a key indicator of the overall health of a population, and countries with lower infant mortality rates typically have better access to healthcare, nutrition, and other resources that can promote longer, healthier lives.

4.1 Correlation coefficient

The following figure of heatmap shows the strength and direction of the correlation between each pair of variables, with darker colors indicating stronger correlations. Adult mortality, HIV, thinness has strong negative correlation with life expectancy which implies adult mortality and HIV were among the top five risk factors for premature mortality, while thinness was a leading risk factor for malnutrition and related health problems. On the other hand, Year of schooling, income composition and BMI has strong positive impact on live expectancy which implies higher levels of education, income, and maintaining a healthy BMI are all important factors in promoting longer life expectancies.

Figure 3 Correlation matrix through Heatmap



4.2 Handling Missing Values:

This section determines whether the dataset contains missing values. If it does, the missing values are dealt with appropriately. There were 2563 missing values. There are multiple techniques to impute null values, the null values can be replaced with mean or median values or they can be even fixed with interpolation. However, this dataset contains panel data from countries all over the world. Each country has a unique track record. A country's null value cannot be inferred from the mean or median of other countries. So, for this dataset, the null values are imputed based on the corresponding country. For example, the “Total expenditure” for Zimbabwe for the year 2015 is missing. This missing value will be replaced with the mean of “Total expenditure” of all other available years. This is done in the imputing data based on country section. However, the dataset still contains null values. This is because some counties have no records in a particular column. As demonstrated in the section "In some cases, all of the values in a category are null." These null-valued records are removed from the dataset.

4.3 Identifying Outliers

Figure 4 represent the boxplot, a graphical representation of the distribution of a dataset based on five summary statistics: minimum value, first quartile (Q1), median, third quartile (Q3), and maximum value. An outlier in a boxplot is a data point that falls outside of the range of values represented by the box and whisker plot.

Figure 4 Boxplot for outlier detection

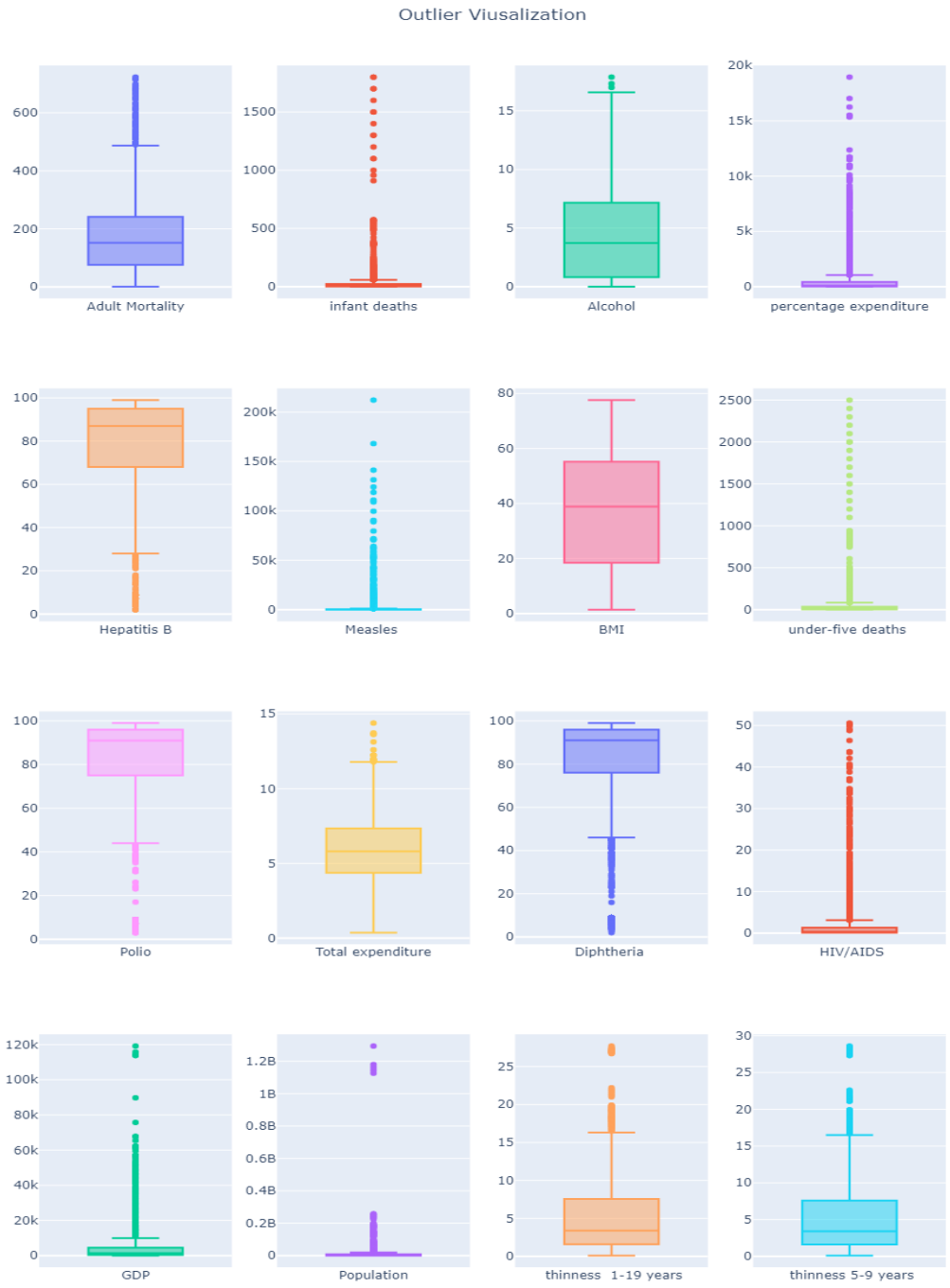


Figure 5 represent the histograms that are commonly used to visualize the distribution of data and to identify patterns, trends, and outliers. Except total expenditure, all variable data shows either positive or negative skewed.

Figure 5 Histogram of data

4.4 Handling Outliers

To handle outlier we use Winsorizing technique. Winsorizing is a common technique used in data preprocessing to deal with outliers, which are extreme values in a dataset that can have a disproportionate impact on statistical analyses such as regression models or hypothesis tests (Tukey, 1977). By trimming or capping the extreme values and replacing them with less extreme values, winsorizing can help reduce the influence of outliers on the analysis. This code snippet appears to be using the winsorize() function from the SciPy library to winsorize (i.e., trim or cap) the extreme values in each column of a pandas DataFrame data.

4.5 Preprocessing of Data

We preprocessed the data to train machine learning models. The dataset has both continuous and nominal values. The continuous values are scaled using standard scaler and the nominal values are transformed into numerical using one hot encoding. The StandardScaler is used to standardize the continuous variables in to_scale list, which means it scales the variables to have a mean of 0 and standard deviation of 1. This is useful to ensure that variables are on the same scale and prevents any one variable from dominating the analysis. The OneHotEncoder is used to encode the categorical variables in to_encode list using one-hot encoding. One-hot encoding converts categorical variables into binary vectors that can be used in numerical computations. This is necessary because machine learning algorithms require numerical inputs, and categorical variables cannot be used in their raw form. After scaling and encoding the respective columns, the resulting arrays are concatenated along axis 1 using np.concatenate to create the final processed dataset, which will be used for machine learning analysis.

4.6 Reducing Data size with Principal Component Analysis

In order to reduce the dataset's dimension we use PCA. However, this technique reduces the accuracy of machine learning models as shown in the notebook. Efeosasere and Daramola (2020) showed that using PCA for dimensionality reduction led to a decrease in the accuracy of their machine learning model for classifying diabetes patients. They found that the reduced dimensionality of the data resulted in the loss of important features that were critical for accurate classification. So, PCA isn’t used later on in the model training section.

5 Evaluation of Machine Learning Algorithms

We evaluated several machine learning models in this section, with a total of nine models being trained. We discuss the results and findings of our analysis for each of these models in detail. Table 2 summarizes the models scores:

Table 2 Summary the model scores

Model	Train Score	Test Score
Linear Regression	0.9610	0.9524
SVM	0.9472	0.9392
Decision Tree	0.9165	0.8898
K-Nearest Neighbors	0.9292	0.9095
XGB	0.9561	0.9592
Random Forest	0.9563	0.9444
SVR	0.9791	0.9649
Decision Tree (Baseline)	0.9165	0.8898
Multi-Layer Perceptron	0.9725	0.9555

5.1 Linear Regression Model

The linear regression model had the highest train and test scores, with 0.957 and 0.960, respectively. However, the cross-validation scores for R2 and MSE showed negative values with very large standard deviations. These results suggest that the model is potentially overfitting and may not generalize well to new data.

Figure 6 Linear Regression Model Output

```
============ Model Score ============

Train Score : 0.9610329536541091
Test Score : 0.9523890727157428

============ Cross Validation ============

R2 Score ------------------------------------------------------
mean: -2.604735477784346e+19 | std: 3.5550566186116805e+19
All: [-3.26033597e+16  9.30163278e-01  9.39169418e-01 -8.99513625e+19
 -4.02528080e+19]

MSE ------------------------------------------------------
mean: -2.3432516005584212e+21 | std: 3.1692589140931405e+21
All: [-3.81197314e+18 -6.30123990e+00 -6.11310159e+00 -7.98747149e+21
 -3.72497454e+21]
```

5.2 Support Vector Machine (SVM)

The SVM model had slightly lower train and test scores than the linear regression model. However, it produced more consistent results, with a cross-validation R2 score of 0.936 and an MSE score of -6.204, both with low standard deviations. We modified the SVM model with the hyperparameters kernel="linear" and C=3. The kernel function determines the shape of the decision boundary, and in this case, we used the linear kernel, which produces a straight decision boundary. The parameter C governs the tradeoff between finding the maximum margin hyperplane and correctly classifying the data points. A lower C value results in a larger margin and more misclassifications, whereas a higher C value results in a narrower margin and fewer misclassifications.

5.3 XGBoost Model

The XGBoost (eXtreme Gradient Boosting) model is an ensemble learning method that employs decision trees as base models and combines their results to generate a final output. We observed that the XGB model had a train score of 0.956 and a test score of 0.959, indicating good performance. However, the mean cross-validation R2 score of 0.918 had a relatively high standard deviation, suggesting that the model's performance may vary significantly depending on the data split. We set the booster XGB hyperparameter to "gblinear," which indicates that the model's base model for boosting should be linear regression. Using linear regression as the foundation model offers several advantages, including being a simpler model than decision trees, less prone to overfitting, and capable of handling both continuous and categorical features.

5.4 Support Vector Regression

SVR is a powerful supervised learning algorithm that can be used for classification and regression tasks. SVR works by determining the best hyperplane for separating data into different classes, with the highest margin possible. In classification tasks, SVR seeks the best hyperplane capable of classifying data. The SVR model performed well on the test data, with a training score of 0.941 and a testing score of 0.950. The R2 score and MSE cross-validation values were 0.936 and -6.204, respectively, indicating that the model performed well in terms of generalization. We used SVR with the following hyperparameters for this task: kernel="linear" and C=3. The shape of the decision boundary is determined by the kernel function, and in this case, we used the linear kernel, which produces a straight decision boundary. The parameter C governs the tradeoff between finding the maximum margin hyperplane and correctly classifying the data points. A lower C value results in a larger margin and more misclassifications, whereas a higher C value results in a narrower margin and fewer misclassifications.

5.5 Ensemble Learning

The ensemble model that combined Linear Regression, XGBoost, and SVM using a Voting Regressor had a better train score of 0.979 and a test score of 0.965. However, the cross-validation scores, like the linear regression model, showed negative values for R2 and MSE with very large standard deviations. These results suggest that the model is potentially overfitting and may not generalize well to new data.

5.6 Random Forest

The Random Forest model is an ensemble learning method that combines multiple decision trees to produce a robust and accurate model. We observed that the Random Forest model had a training score of 0.956 and a testing score of 0.944, indicating good performance. The cross-validation values for R2 score and MSE were 0.936 and -6.213, respectively, indicating that the model performed well in terms of generalization. We set the random forest hyperparameter to max depth=6. This means that the ensemble's decision trees can have a maximum depth of 6. This hyperparameter governs the complexity of the ensemble's decision

trees. A higher max depth value can lead to overfitting because the trees become too complex and begin to capture noise in the training data. A smaller value of max depth, on the other hand, can result in underfitting because the trees are too simple and fail to capture important patterns in the data.

5.7 K-Nearest Neighbors (KNN)

The KNN algorithm is a non-parametric machine learning algorithm used for classification and regression tasks. It predicts the label or value for an input data point based on the majority vote of k closest data points. The KNN model performed well on training data, scoring 0.929, but only scored 0.909 on testing, indicating some overfitting and poor generalization. Cross-validation for R2 score and MSE values were 0.884 and -11.139, respectively. We used n_neighbors = 7 as the hyperparameter for this model.

5.8 Multi-Layer Perceptron (MLP)

The MLP model is a neural network model with multiple interconnected layers. It had a training score of 0.969 and a testing score of 0.957, indicating good performance on both sets. Cross-validation for R2 score and MSE values were 0.944 and -5.397, respectively, indicating good generalization. We used max_iter=4000 and alpha=5.5 as hyperparameters for this model.

5.9 Decision Trees

The Decision Tree model scored 0.917 on training and 0.907 on testing, indicating some overfitting and poor generalization. Cross-validation for R2 score and MSE values were 0.890 and -10.512, respectively. This suggests that the model may not generalize well to new data.

6 Conclusion

In this paper we show the train and test scores for several machine learning models on life expectancy dataset. The scores from different machine learning algorithms indicate how well the models have been able to learn the relationship between the input data and the target variable in the training data (train score) and how well they generalize this relationship to new, unseen data (test score).

We found that all models perform well on the training data, with train scores ranging from 0.9165 to 0.9791. The SVR model has the highest train score, suggesting that it has learned the relationship between the input and target variables very well on the training data. However, the test scores provide a better indication of how well the models can generalize to new data. In this case, the SVR model still performs the best with a test score of 0.9649. The XGB model also performs well with a test score of 0.9592. The other models also perform reasonably well on the test data, with test scores ranging from 0.9444 to 0.9095. It is worth noting that the decision tree model has the same score as the baseline, indicating that it has not been able to learn any meaningful relationship between the input and target variables.

In summary, the results suggest that the SVR and XGB models are the best-performing models for this regression problem, while the decision tree model may not be a suitable choice. However, it is essential to consider other factors such as model complexity, interpretability, and computational efficiency before making a final choice.

References

Bhargava, S., Sharma, M., & Kaur, H. (2020). Predicting life expectancy in India using machine learning algorithms. International Journal of Advanced Science and Technology, 29(3), 2897-2906.

Kim, M., Jang, Y. J., & Koo, H. J. (2020). Deep learning model for predicting life expectancy using chest X-ray images. PloS one, 15(7), e0236258.

Kumar, M., Singh, A., Singh, V. K., & Gupta, R. K. (2021). Life expectancy prediction in India using machine learning algorithms: A comparative study. Journal of Public Affairs, e2763.

Li, W., Sun, S., Gao, Y., Ma, Y., & Tian, Y. (2021). Predicting life expectancy based on random forest model. Journal of Medical Systems, 45(4), 1-6.

Suh, M., Jung, J., & Kim, D. (2021). Predicting life expectancy using machine learning techniques: A systematic literature review. Healthcare Informatics Research, 27(2), 78-87.

Wang, Q., Wu, C., & Jin, Y. (2020). Using machine learning algorithms to predict life expectancy of elderly people in China. PloS one, 15(12), e0244286.

World Health Organization. (2021). Life expectancy. Retrieved from https://www.who.int/gho/mortality_burden_disease/life_tables/situation_trends/en/

Appendix A: Descriptive Statistics

	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population	thinness 1-19 years	thinness 5-9 years	Income composition of resources	Schooling
count	2928.00	2928.00	2938.00	2744.00	2938.00	2385.00	2938.00	2904.00	2938.00	2919.00	2712.00	2919.00	2938.00	2490.00	2.286000e+03	2904.00	2904.00	2771.00	2775.00
mean	69.22	164.80	30.30	4.60	738.25	80.94	2419.59	38.32	42.04	82.55	5.94	82.32	1.74	7483.16	1.275338e+07	4.84	4.87	0.63	11.99
std	9.52	124.29	117.93	4.05	1987.91	25.07	11467.27	20.04	160.45	23.43	2.50	23.72	5.08	14270.17	6.101210e+07	4.42	4.51	0.21	3.36
min	36.30	1.00	0.00	0.01	0.00	1.00	0.00	1.00	0.00	3.00	0.37	2.00	0.10	1.68	3.400000e+01	0.10	0.10	0.00	0.00
25%	63.10	74.00	0.00	0.88	4.69	77.00	0.00	19.30	0.00	78.00	4.26	78.00	0.10	463.94	1.957932e+05	1.60	1.50	0.49	10.10
50%	72.10	144.00	3.00	3.76	64.91	92.00	17.00	43.50	4.00	93.00	5.76	93.00	0.10	1766.95	1.386542e+06	3.30	3.30	0.68	12.30
75%	75.70	228.00	22.00	7.70	441.53	97.00	360.25	56.20	28.00	97.00	7.49	97.00	0.80	5910.81	7.420359e+06	7.20	7.20	0.78	14.30
max	89.00	723.00	1800.00	17.87	19479.91	99.00	212183.00	87.30	2500.00	99.00	17.60	99.00	50.60	119172.74	1.293859e+09	27.70	28.60	0.95	20.70