Life Expectancy Prediction

CSCI B-565 Data Mining
Final Project Report

By

Laxmi Soujanya Taduru - ltaduru@iu.edu
Srichandana Chakilam - schakil@iu.edu

Under the Supervision of

Dr. Yuzhen Ye

Chair, Computer Science
Professor of Informatics and Computer Science
Indiana University Bloomington

ABSTRACT

Life expectancy gauges the overall health of a community, thereby giving the scope of improving several factors that affect it. In the past few years, there has been a huge improvement in the health sector which has impacted the average age of death of the population. This also resulted in the overall deterioration of the human mortality rate, especially in the developing countries.

However, the factors affecting this were not completely analyzed. The effect of alcohol consumption, adult mortality rate, immunization like Hepatitis, Diphtheria and Measles, GDP of a country, and literacy rate have a larger role than expected.

The goal of this project is to predict life expectancy based on different factors like immunization, mortality, economic and other health related factors.

Keywords: Dataset, Features, Preprocessing, Data Visualization, Target Variable, Models, Accuracy score, r2 score, Linear Regressor, Ridge Regressor, Decision Tree Regressor, AdaBoost Regressor, XGBoost Regressor.

TABLE OF CONTENTS

1. INTRODUCTION

Many studies in the past have taken multiple factors into consideration to understand and analyze the life expectancy of people in a country but overlooked the importance of human development index, immunization, mortality, economic and other health related factors.

This project contributes to analyze and understand the factors that influence the life span of the population of several countries and proposing different algorithms and comparing their efficiency and performance in a Kaggle dataset. Given its predictor variables, the models implemented in the project also predict the life expectancy values.

1.1 Dataset:

The dataset used in the project has been made available by World Health Organization (WHO) for health analysis purposes. The data contains information related to life-expectancy, and its factors that has been collected from 193 countries and maintained in WHO repository website. The corresponding economic data has been collected from United Nations website. It contains data from the year 2000-2015.

This dataset is available on Kaggle. It consists of 2938 items and 22 columns, with ‘life expectancy’ being the target variable and 21 predictor variables.

2. METHODS

The components of the project are:

1. Data Analysis
2. Data Preprocessing
3. Data Visualization
4. Model implementation

2.1 Data Analysis:

After loading the data, it is necessary to understand the structure and features of the data. Hence, we performed data analysis as the first step in this project. In this critical step, simple operations have been implemented on the data. For instance, the method – info() provides details about the features' datatypes, count of non-null columns alongside feature names as shown below.

```
In [502]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Country                          2938 non-null   object
 1   Year                             2938 non-null   int64
 2   Status                           2938 non-null   object
 3   Life expectancy                  2928 non-null   float64
 4   Adult Mortality                  2928 non-null   float64
 5   infant deaths                    2938 non-null   int64
 6   Alcohol                          2744 non-null   float64
 7   percentage expenditure           2938 non-null   float64
 8   Hepatitis B                      2385 non-null   float64
 9   Measles                          2938 non-null   int64
 10   BMI                             2904 non-null   float64
 11  under-five deaths                2938 non-null   int64
 12  Polio                            2919 non-null   float64
 13  Total expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15   HIV/AIDS                        2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18   thinness  1-19 years            2904 non-null   float64
 19   thinness 5-9 years              2904 non-null   float64
 20  Income composition of resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

Figure 1: Dataset Information

We have also identified the missing values in the dataset.

```
In [509]: total = df.isnull().sum()
          missing_values = pd.concat([total], axis=1, keys=['Total'])
          missing_values
```

Status	0
Life expectancy	10
Adult Mortality	10
infant deaths	0
Alcohol	194
percentage expenditure	0
Hepatitis B	553
Measles	0
BMI	34
under-five deaths	0
Polio	19
Total expenditure	226
Diphtheria	19

Figure 2: Missing values per each column

2.2 Data Preprocessing:

In this step, the raw dataset will be transformed into the data that can be fed into the models we implement. Some of the important preprocessing steps include:

1. Changing column names for easy access by using underscores and removing extra spaces.

```
df.rename(columns = {" BMI " :"bmi", "Polio": "polio",
                                 "Life expectancy ": "life_expectancy",
                                 "Adult Mortality":"adult_mortality",
                                 "infant deaths":"infant_deaths",
                                 "percentage expenditure":"percentage_expenditure",
                                 "Hepatitis B":"hepatitisB",
                                 "Alcohol":"alcohol",
                                 "Status":"status",
                                 "Measles ":"measles",
                                 "under-five deaths ": "under_five_deaths",
                                 "Total expenditure":"total_expenditure",
                                 "Diphtheria ": "diphtheria",
                                 "Population" : "population",
                                 " thinness  1-19 years":"thinness_1-19_years",
                                 " thinness 5-9 years":"thinness_5-9_years",
                                 " HIV/AIDS":"HIV/AIDS",
                                 "Income composition of resources":"income_composition_of_resources",
                     "Schooling":"schooling"}, inplace = True)
```

	Country	Year	status	life_expectancy	adult_mortality	infant_deaths	alcohol	percentage_expenditure	hepatitisB	measles	...	polio	total_expendi
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.000000	68.0	31	...	67.0	
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.000000	7.0	998	...	7.0	
2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.000000	73.0	304	...	73.0	
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.000000	76.0	529	...	76.0	
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.000000	79.0	1483	...	78.0	

2938 rows × 22 columns

Figure 3.2.1: Renaming columns

2. Filling the null values that we identified during EDA, with the mean values of its corresponding attribute.

```
df.isnull().sum()
```

```
Country                         0
Year                            0
status                          0
life_expectancy                 0
adult_mortality                 0
infant_deaths                   0
alcohol                         0
percentage_expenditure          0
hepatitisB                      0
measles                         0
bmi                             0
under_five_deaths               0
polio                           0
total_expenditure               0
diphtheria                      0
HIV/AIDS                        0
GDP                             0
population                      0
thinness_1-19_years             0
thinness_5-9_years              0
income_composition_of_resources 0
schooling                       0
dtype: int64
```

Figure 3.2.2: Columns showing no null values after filling with mean values.

3. Converting the categorical variables into numerical variables using encoding techniques.

- There are two object-based attributes (‘Status’, ‘Country’) in the dataset. To apply the training models, these attributes must be handled.
- The column- *‘Status’* is a categorical attribute with only two values (‘Developing’ and ‘Developed’), it can be replaced with values ‘0’ and ‘1’.
- The column - *‘Country’* is a nominal attribute with 192 values, it can be handled using Feature Encoding; It converts categorical data into dummy or indicator variables that can be used to feed the regressor models.
- A similar manipulation technique has been used on the column - *‘Year’*.

```
df
```

	status	life_expectancy	adult_mortality	infant_deaths	alcohol	percentage_expenditure	hepatitisB	measles	bmi	under_five_deaths	...	Country_United Republic of Tanzania
0	0	65.0	263.0	62	0.01	71.279624	65.0	1154	19.1	83	...	0
1	0	59.9	271.0	64	0.01	73.523582	62.0	492	18.6	86	...	0
2	0	59.9	268.0	66	0.01	73.219243	64.0	430	18.1	89	...	0
3	0	59.5	272.0	69	0.01	78.184215	67.0	2787	17.6	93	...	0
4	0	59.2	275.0	71	0.01	7.097109	68.0	3013	17.2	97	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
2933	0	44.3	723.0	27	4.36	0.000000	68.0	31	27.1	42	...	0
2934	0	44.5	715.0	26	4.06	0.000000	7.0	998	26.7	41	...	0
2935	0	44.8	73.0	25	4.43	0.000000	73.0	304	26.3	40	...	0
2936	0	45.3	686.0	25	1.72	0.000000	76.0	529	25.9	39	...	0
2937	0	46.0	665.0	24	1.68	0.000000	79.0	1483	25.5	39	...	0

2938 rows × 229 columns

Figure 3.2.3: Categorical columns after being encoded into numerical variables.

4. Normalizing the data.
    - The scale of data for some features may be significantly different from those of others, which may harm the performance of our models.
    - Using Min-max scaling, we can limit the data values of a column to a specific range using each column’s minimum and maximum value.
    - This has been applied to all the numeric attributes in our dataset.

```
from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler(feature_range =(0, 1))
df[numeric_features] = min_max_scaler.fit_transform(df[numeric_features])
print(df[:1])
```

```
   status  life_expectancy  adult_mortality  infant_deaths  alcohol  \
0       0             65.0         0.362881       0.034444      0.0

   percentage_expenditure  hepatitisB   measles       bmi  under_five_deaths  \
0                0.003659    0.653061  0.005439  0.209733             0.0332

   ...  Country_United Republic of Tanzania  Country_United States of America  \
0  ...                                    0                                 0

   Country_Uruguay  Country_Uzbekistan  Country_Vanuatu  \
0                0                   0                0

   Country_Venezuela (Bolivarian Republic of)  Country_Viet Nam  \
0                                           0                 0

   Country_Yemen  Country_Zambia  Country_Zimbabwe
0              0               0                 0

[1 rows x 229 columns]
```

Figure 3.2.4: Normalization of all numerical features in the dataset.

2.3 Data Visualization:

During this step, the preprocessed or formatted data is represented using visual elements like bar plots, histograms etc. A few observations have been made in the project

1. Using correlation analysis, the variables that impact the predictor variable can be found.
2. Life expectancy has positive correlation with schooling and Income composition of resources.
3. Life expectancy has negative correlation with adult mortality and HIV/AIDS.

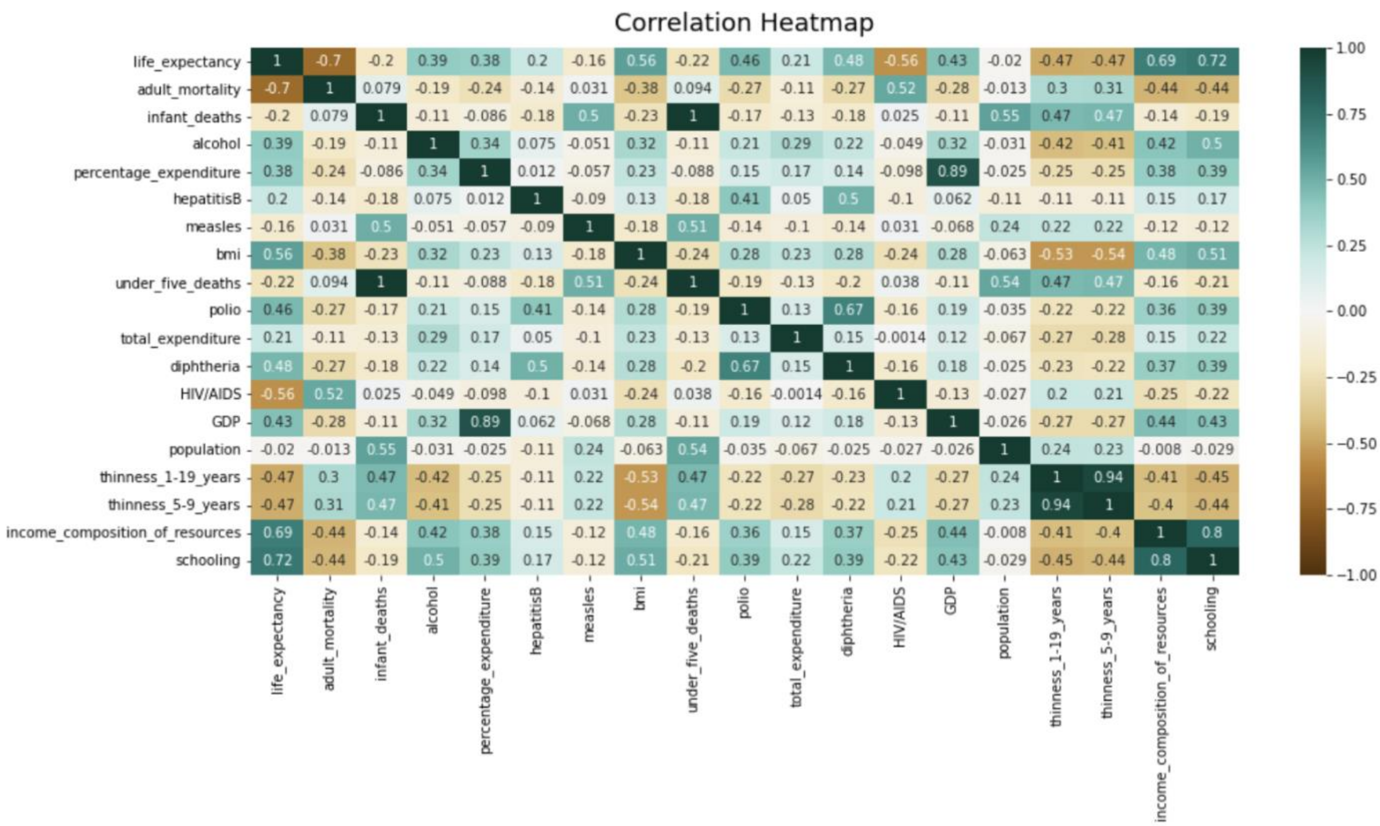

Figure 3.3.1: Correlation heatmap of the dataset.

4. The features like schooling and Income composition of resources, adult mortality, HIV/AIDS may be good predictors of Life Expectancy.

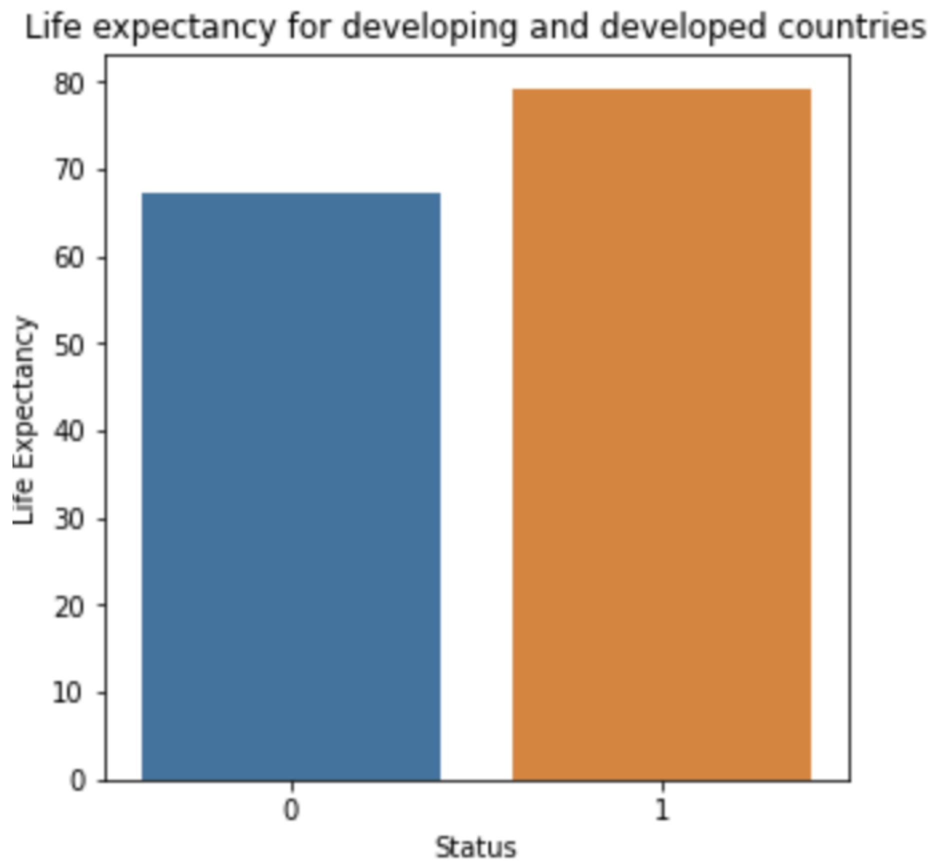

Figure 3.3.2: Bar plot showing the life expectancy for developing and developed countries.

2.4 Model Implementation:

Regression is a technique for determining the relationship between independent variables or characteristics and a dependent variable or result. Once the link between the independent and dependent variables has been estimated, outcomes can be predicted. Following models have been implemented in this project:

1. Linear Regression
2. Ridge Regression
3. Decision Tree Regression
4. AdaBoost – Adaptive Boosting
5. XGBoost – Extreme Gradient Boosting

2.4.1 Linear Regression:

Linear Regression is used to model the relationship between predictor and target variables. The target variable must be continuous in order to apply this model. Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is an explanatory variable, and the other is a dependent variable.

2.4.2 Ridge Regression:

Ridge regression can be used when there is multicollinearity in the dataset or when the number of predictor variables is higher than the number of observations. This method performs L2 regularization.

2.4.3 Decision Tree Regression:

Numerical values can be handled by decision trees. It gradually builds an associated decision tree while dividing a dataset into smaller and smaller sections. This model regresses the data using true or false answers to certain questions.

2.4.4 AdaBoost Regression:

AdaBoost stands for Adaptive boosting, is a meta-algorithm, which enables performance enhancement when combined with other algorithms. It starts by fitting one regressor on the original dataset, and then it fits subsequent copies of the regressor on the same dataset with the weights of the instances being changed in accordance with the error of the most recent prediction. As a result, successive regressors concentrate more on challenging cases.

2.4.5 XGBoost Regression:

XGBoost stands for “Extreme Gradient Boosting”. It can be used directly for **regression predictive modeling**. The model depends on the primary learners which are not good at the remainder. But when all the predictions are combined, they sum up to form final good predictions.

3. RESULTS

The models that are implemented in the model are ranked based on their r2 and accuracy scores. Below are the training and testing scores of each model. R2 score is an accuracy evaluation technique that represents the proportion of the variance for a dependent variable which is determined by an independent variable(s) in a regression model.

3.1 Linear Regression:

```
print("Training Score for Linear Regression:", LRTrainScore)
print("Testing Score for Linear Regression:",LRTestScore)
```

```
Training Score for Linear Regression: 96.44415823252547
Testing Score for Linear Regression: 95.85513589797138
```

Figure 3.5.1: Train and test accuracy scores of Linear Regressor model.

Overall, the accuracy for the model is 95.85% and can predict the life expectancy value for a given input.

```
y_pred=model_LR.predict(X_test)
LR_r2_score=r2_score(y_test,y_pred)
print("R2 Score for Linear Regression:", LR_r2_score)
```

```
R2 Score for Linear Regression: 0.9585513589797138
```

Figure 3.5.1.1: R2 score for Linear Regressor model.

Similarly, the evaluated r2 metric is 0.95 for this model.

3.2 Ridge Regression:

```
print("Training Score for Ridge Regression:",RRTrainScore)
print("Testing Score for Ridge Regression:",RRTestScore)
```

```
Training Score for Ridge Regression: 96.34635349641965
Testing Score for Ridge Regression: 95.73920697195675
```

Figure 3.5.2: Train and test accuracy scores of Ridge Regressor model.

The accuracy of the Ridge regressor for the given dataset is 95.73% and can predict the life expectancy value for a given input.

```
y_pred=ridge_regressor.predict(X_test)
RR_r2_score=r2_score(y_test,y_pred)
print("R2 Score for Ridge Regression:", RR_r2_score)
```

```
R2 Score for Ridge Regression: 0.9573920697195675
```

Figure 3.5.2.1: R2 score for Ridge Regressor model.

The evaluated r2 metric is 0.95 for Ridge regressor model.

3.3 Decision Tree Regression:

```
print("Training Score for Decision Tree Regression:",DTRTrainScore)
print("Testing Score for Decision Tree Regression:",DTRTestScore )
```

```
Training Score for Decision Tree Regression: 98.7245769722907
Testing Score for Decision Tree Regression: 93.11880644626416
```

Figure 3.4.3: Train and test accuracy scores of Decision Tree Regressor model.

The accuracy of the Decision Tree regressor for the given dataset is 93.11%.

```
y_pred=dt.predict(X_test)
DTR_r2_score=r2_score(y_test,y_pred)
print("R2 Score for Decision Tree Regression:", DTR_r2_score)
```

```
R2 Score for Decision Tree Regression: 0.9311880644626416
```

Figure 3.5.3.1: R2 score for Decision Tree Regressor model.

The evaluated r2 metric is 0.93 for Decision Tree regressor model.

3.4 AdaBoost Regression:

```
print("Training Score for AdaBoost Regression:",ABRTrainScore)
print("Testing Score for AdaBoost Regression:",ABRTestScore )
```

```
Training Score for AdaBoost Regression: 90.41141508111417
Testing Score for AdaBoost Regression: 89.32162620975603
```

Figure 3.5.4: Train and test accuracy scores of AdaBoost Regressor model.

The accuracy of the AdaBoost regressor for the given dataset is 89.32%.

```
y_pred=ada_reg.predict(X_test)
ABR_r2_score=r2_score(y_test,y_pred)
print("R2 Score for AdaBoost Regression:", ABR_r2_score)
```

```
R2 Score for AdaBoost Regression: 0.8932162620975603
```

Figure 3.5.4.1: R2 score for AdaBoost Regressor model.

The evaluated r2 metric is 0.89 for AdaBoost regressor model.

3.5 XGBoost Regression:

```
print("Training Score for XGBoost Regression:",XGBTrainScore)
print("Testing Score for XGBoost Regression:",XGBTestScore )
```

```
Training Score for XGBoost Regression: 91.69090962456157
Testing Score for XGBoost Regression: 90.36774014797683
```

Figure 3.5.5: Train and test accuracy scores of XGBoost Regressor model.

The accuracy of the XGBoost regressor for the given dataset is 90.36%.

```
y_pred=xgb_r.predict(X_test)
XGB_r2_score=r2_score(y_test,y_pred)
print("R2 Score for XGBoost Regression:", XGB_r2_score)
```

```
R2 Score for XGBoost Regression: 0.9036774014797683
```

Figure 3.5.5.1: R2 score for XGBoost Regressor model.

The evaluated r2 accuracy score is 0.90 for XGBoost regressor model.

```
FinalScore
```

	Model name	Accuracy score(testing)
0	Linear Regression	95.855136
1	Ridge Regression	95.739207
2	Decision Tree Regression	93.118806
3	Adaboost Regression	89.321626
4	XGBoost Regression	90.367740

Figure 3.5.6: Test accuracy scores of all models.

```
FinalR2Score
```

	Model name	R2 score(testing)
0	Linear Regression	0.958551
1	Ridge Regression	0.957392
2	Decision Tree Regression	0.931188
3	Adaboost Regression	0.893216
4	XGBoost Regression	0.903677

Figure 3.5.7: R2 Test accuracy scores of all models.

The above table of accuracy scores help us understand the performance of individual models when test data is applied.

4. DISCUSSION

Data Mining modules implemented in the project:
Exploratory Data Analysis: As part of this step, information from the dataset has obtained with help of various visualizations, summary statistics etc.

Feature engineering: Meaningful insights have been derived from raw data and it has been manipulated in a way to improve model's performance.

Model Implementation: Various models have been implemented in this project for training the model a set of training data has been fed into it with help of which unknown data can be used to evaluate which can help us make better decisions.

Data limitations: Our dataset consisted of roughly 3000 items which is a fair amount of data according to health sector perspective. However, it is not considered to be a lot of data for training regressor models. Although we have tried to split the data into test and train data by 20% and 80% respectively, more data would have been desirable for training the model, in order to increase the accuracy and reduce overfitting.

Another challenge is that there are some outliers present in the dataset. Since the size of the data is not very large, these outliers were not removed, which would adversely affect while training the models and hamper the performance. It is also possible that these outliers are not completely noisy to be eliminated.

Interpretation of the output: As part of this project, we have performed exploratory data analysis, pre-processed the data for it to be trainable to models and visualized it for analysis. Although the models – AdaBoost and XGBoost are extremely powerful and outperform the other models, the Linear regressor model and Ridge regressor works better as the current dataset is not large.

Conclusions: We choose to understand the best model that predicts the output target variable – ‘Life expectancy’ when different predictor variable values are fed into each model. All the models implemented in this project seem to predict the life expectancy given its independent variables. Using evaluating criteria, we could clearly see better performance being shown by Ridge and Linear regressors competing, albeit the presence of powerful regressors like Adaptive Boosting or XGBoost.

5. REFERENCES

- WHO statistical information system [online database]. Life expectancy at birth. Geneva: World Health Organization; 2016 (https://www.who.int/whosis/whostat2006DefinitionsAndMetadata.pdf, accessed 4 January 2016).

- Global Health Observatory data repository. World Health Statistics [online database]. Geneva: World Health Organization; 2013 (https://apps.who.int/gho/data/node.main.1?lang=en, accessed 4 January 2016).

- World Bank. Data [online database] Washington (DC): World Bank; 2013 (https://databank.worldbank.org/source/world-development-indicators, accessed 27 June 2019).

- Klenk J, Keil U, Jaensch A, Christiansen MC, Nagel G. Changes in life expectancy 1950–2010: contributions from age-and disease-specific mortality in selected countries. Popul Health Metr. 2016;14(1):20–30.

- Ho JY, Hendi AS. Recent trends in life expectancy across high income countries: retrospective observational study. BMJ. 2018;362. pmid:30111634

- Kinsella K, Velkoff V. Life expectancy and changing mortality. Aging Clin Exp Res. 2002;5(14):322–32

- Sinkon Nayak, Manjusha Pandey, Siddharth S. Rautaray. A Proposal for Life Expectancy Analysis using Machine Learning Techniques (https://www.researchgate.net/publication/365662993_A_Proposal_for_Life_Expectancy_Analysis_using_Machine_Learning_Techniques)

- https://www.researchgate.net/publication/356235215_Life_Expectancy_Prediction_Analysis_using_ML

- Dataset - https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who

- AdaBoost Regression - https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

- Linear Regression - https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

- Decision Tree Regression - https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

- Ridge Regression - https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

- XGBoost Regression - https://xgboost.readthedocs.io/en/stable/python/python_api.html

- Feature Engineering - https://www.projectpro.io/article/8-feature-engineering-techniques-for-machine-learning/423