**Machine Learning Solution for Life Expectancy Prediction**

***Full name: Chi Nguyen Chau***

***SID: 22156675***

## Frame the problem

**Life expectancy** has several attributes such as *adult mortality, infant deaths ratio, alcohol consumption, the percentage expenditure of GDP per capita on health, Hepatitis B, Polio, Diphtheria immunization coverages among 1-year-olds* *and so on*. Building a model to predict this value can provide a tool for governments to evaluate how efficient their policies in improving the citizen’s ages are and then make appropriate adjustments. It is a supervised learning and multivariate regression task because the labelled training examples are given, the model uses multiple features to make predictions and the outputs are expected as specific numeric values instead of the classification. For this model, plain batch learning is enough due to no continuous flow of data, no need to adjust for rapidly changing data and a small dataset able to fit into machine’s memory. The root mean square error (RMSE) is the measure chosen to evaluate our model’s performance.

## Get and Explore the Data

The “Life\_Expectancy\_Data.csv” dataset includes 2938 rows and 22 columns with 17 columns containing missing values which are needed to be dealt with before feeding the model.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Exploring the correlation between Life expectancy and other columns, we can see that Population doesn’t have significant correlation with life expectancy, so we can get rid of it from our training and testing data.

A screenshot of a computer

Description automatically generated

When it comes to the correlation among features, the following heatmap depicts that infant deaths and under-five deaths, GDP and percentage expenditure, thinness 1-19 years and thinness 5-9 years, income composition of resources and schooling are pairs that exist high correlation.

A white background with text

Description automatically generated

A screenshot of a computer screen

Description automatically generated

These findings provide some ideas for the transformation pipeline such as filling missing values, feature removal and feature combination. Before moving to the data preparation step, we create a label list ***y*** which represents life expectancy and an attribute matrix ***X*** which represents the rest of columns in the given dataset. We also add a ***LE class*** column that is the classification of label ***y*** calculated by rounding down the fraction of ***y*** over 10. ***X*** and ***y*** then are split into a train set and a test set with the ratio of 80% and 20% respectively. The final sets we have include X\_train, X\_test, y\_train, y\_test, y\_train\_log and y\_test\_log (used for LogisticRegression).

## Preparing the Data for Machine Learning Algorithm

For convenience in applying the transformation process to new instances, we create a pipeline that can handle both numerical and categorical features.

In numerical pipeline, there are four steps including replacing null value by the mean of that attribute corresponding to the country, selecting the attributes to transfer to next steps, combining the thinness 1-19 years and thinness 5-9 years by adding their values together and rescaling the attributes by StandardScaler.

A computer screen shot of a computer code

Description automatically generated

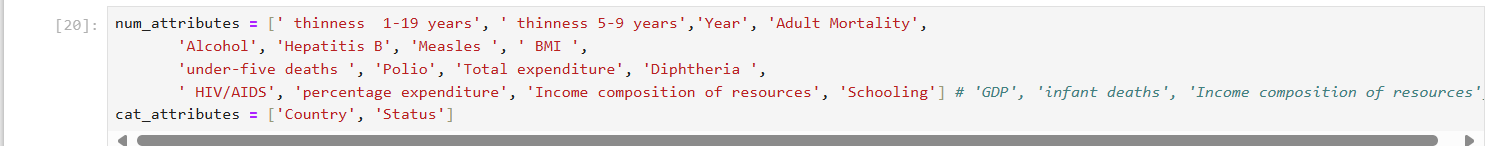
A white background with a black and white flag

Description automatically generated with medium confidence

A close-up of a computer screen

Description automatically generated

As mentioned above, we don’t select the feature *population* due to lack of correlation to the label. In addition, we also don’t pick the *GDP* and *infant deaths* because they have strong correlation to *the percentage expenditure* and *under-five deaths* respectively. The empirical results suggest that removing these features leads to a lower RMSE. We keep *the income composition of resources* and *schooling* due to their high correlation to the label and the higher RMSE when getting rid of them.



|  |  |  |
| --- | --- | --- |
| GDP + percentage | GDP | Percentage expenditure |
| A screenshot of a computer code  Description automatically generated | A screenshot of a computer code  Description automatically generated | A screenshot of a computer code  Description automatically generated |
| Infant deaths + Under-five deaths | Infant deaths | Under-five deaths |
| A screenshot of a computer code  Description automatically generated | A screenshot of a computer code  Description automatically generated | A screenshot of a computer code  Description automatically generated |
| Schooling + Income composition of resources | Schooling | Income composition of resources |
| A white background with black text  Description automatically generated | A screenshot of a computer  Description automatically generated | A white screen with black text  Description automatically generated |

In categorical pipeline, the OneHotEncoder method is utilized to convert the *Country* and *Status* features into numerical sparse matrix because regressors accept numeric only.

The implementation of these pipelines are as follows:

A screenshot of a computer

Description automatically generated

## Select and Train a Model

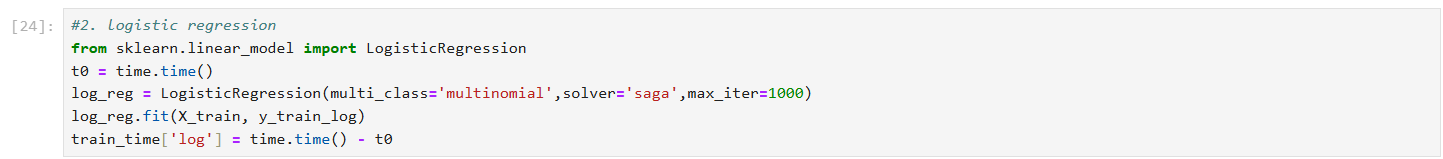
In order to find the most suitable model, the idea was that we will try training several models and evaluate their performances as well as measure the training and testing time. Based on these values, we will suggest which model is indeed fit enough.

The models were trained include:

* LinearRegression



* LogisticRegression



* Polynomial Regression

A white background with black and white text

Description automatically generated with medium confidence

* Support Vector Machine

A white background with colorful text

Description automatically generated

* DecisionTreeRegressor

A close-up of a computer screen

Description automatically generated

* MLPRegressor

A white screen with colorful text

Description automatically generated

We also used some ensemble learning methods to enhance the performance such as:

* RandomForestRegressor

A white background with colorful text

Description automatically generated

* GradientBoostingRegressor (GBR)

A group of colorful text

Description automatically generated

* VotingRegressor

A white background with blue and brown dots

Description automatically generated

In addition, GridSearchCV was the fine-tuning method applied to our models to find the optimal hyper-parameters in random given collections.

The performance was measured as follows:

* RMSE:

A white text with black text

Description automatically generated

The output of LogisticRegression is classification so it isn’t suitable for our purpose. For the remaining models, the performance results show that MLP achieved a low RMSE (1.70) and the ensemble learning with Voting Regressor that voted between all models enhanced the performance significantly (RMSE = 1.64). Considering that the RMSE score of Polynomial Regression is much higher than the others, we can propose a hypothesis that it isn’t an apt model for our prediction. We tested this hypothesis by getting rid of the poly regressor from the Voting Regressor.

A close-up of a flag

Description automatically generated

The scores show that the hypothesis is reasonable, and this removal can improve the performance.

A white text with black text

Description automatically generated

Another hypothesis is that voting without the Decision Tree can achieve a better score because the Voting Regressor has contained the Random Forests already. However, the testing result indicates that Decision Tree contributed significantly to enhancing the performance.

A person holding a blue object

Description automatically generated with medium confidence

A white text with black text

Description automatically generated

* Running time:

A screenshot of a computer

Description automatically generated

Although the MLP and voting models took a long time to train, their predicting time is not significant. This implies that after training, the final model can be used for future predictions.

## Conclusion

According to the final RSME scores, the Voting Regressor that votes between linear, SVM, decision tree, MLP, random forest, GBR achieves the lowest score. The cross validation in train set also provides an approximate score with test set which indicates that it is not overfitting. Therefore, we suggest this model for the life expectancy prediction.

A number on a white background

Description automatically generated