Life Expectancy Prediction

EE 660 Course Project

Project Type: (1) Design a system based on real-world data

Number of student authors: 1

Name:Pin-Hsuan Lee
Email:pinhsual@usc.edu

Date:12/04/2019

1. Abstract

In the project, we need to deal with a real-world dataset from the year 2000-2015 for 193 countries given by WHO to predict life expectancy. There are 2938 points and 20 features including categorical and numerical features. At first, we separate the whole dataset in the proportion 2:8 to test set and training set. Then, separate training dataset in the proportion 2:8 to validation set and new training dataset. In addition to dealing with the categorical features, I also tried to address the outliers and missing data problems to see if we can get better accuracy. After preprocessing, I standardized the feature and applied the correlation method to see the importance of all features to reduce the dimensionality. Then, we used the validation dataset to see MSE for different models, such as random forest for regression, linear regression without or with different regularizations. Then, pick the one that has the lowest MSE value as our optimal model. In the end, use the training model to train the best model with the optimal parameter and apply test data to see the performance of it.

2. Introduction

2.1. Problem Type, Statement and Goals

In this project, I predicted life expectancy belonging to regression problem from real-world dataset. There are some challenges that I need to solve, including:

(1) Categorical features:
There are two kinds of categorical features, nominal and ordinal. For ordinal features, it's easier to solve that we can just assign different values to the features according to the characteristics, like assign 1,2,3 to the size S, M, L. For nominal features, we can apply one hot encoding method. However, we need to be careful since it may increase the dimensionality of the model which can cause overfitting.

(2) High dimensionality of feature space:
In real-world data, the output is influenced by multiple reasons. However, if we train the model with all of the features, it may cause the

curse of dimensionality. Therefore, we need to apply some techniques to reduce the dimensionality.

(3) Outliers:
Outliers are important since depending on how different they are might disproportionately bias the results of a statistical analysis of the data set. Therefore, it is important to identify outliers and evaluate how to treat them.

(4) Nonlinear behaviors:
Since the output is affected by multiple factors, in most situations, the model contains nonlinear behaviors. We will use different models to see which model can fit the data better and pick up the one having the minimum MSE as our final model.

(5) Limited number of training samples:
In the dataset, we only have 2938 points in total and we need to separate these data points to training samples and test samples. Since the data points are limited, missing data handling plays an important role in the whole process.

2.2. Our Prior and Related Work ----- None

2.3. Overview of Our Approach

We compare MSE in different algorithms with different parameters, such as linear regression without/ with different regularizations, and random forest regression with different numbers of dimensions or trees. Then, find the one with the lowest MSE as our final model.

3. Implementation

3.1. Data Set

In the dataset, there are 19 numeric features, 2 categorical features and 1 numerical output.

Feature Name	Type	Description
Input Features		
Country	C	There are 183 countries in total.
Year	N	Year from 2000 to 2015.
Status	C	Developed (17%) or Developing (83%) status.
Adult Mortality	N	Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)
Infant deaths	N	Number of Infant Deaths per 1000 population
Alcohol	N	Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)

Expenditure %	N	Expenditure on health as a percentage of Gross Domestic Product per capita(%)
Hepatitis B	N	Hepatitis B immunization coverage among 1-year-olds (%)
Measles	N	Number of reported Measles cases per 1000 population
BMI	N	Average Body Mass Index of entire population
Under-five deaths	N	Number of under-five deaths per 1000 population
Polio	N	Polio (Pol3) immunization coverage among 1-year-olds (%)
Total Expenditure	N	General government expenditure on health as a percentage of total government expenditure (%)
Diphtheria	N	Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)
HIV/AIDS	N	Deaths per 1000 live births HIV/AIDS (0-4 years)
GDP	N	Gross Domestic Product per capita (in USD)
Population	N	Population of the country
Thinness 1-19 years	N	Prevalence of thinness among children and adolescents for Age 10 to 19 (%)
Thinness 5-9 years	N	Prevalence of thinness among children for Age 5 to 9(%)
Income composition of resources	N	Human Development Index in terms of income composition of resources (index ranging from 0 to 1)
Schooling	N	Number of years of Schooling(years)
Output		
Life Expectancy	N	Life Expectancy in age

(Type: C=> Categorical; N=> Numerical)

3.2. Preprocessing, Feature Extraction, Dimensionality Adjustment

3.2.1. Point has no output values

There are 10 data points have no value in the output “Life Expectancy”. Since it does not stand a large proportion in the dataset, we don’t want to spend a lot of time dealing with it and decide to remove the data having no output value directly.

3.2.2. Categorical features

There are 2 categorical features in the dataset: “Country” and “Status”.

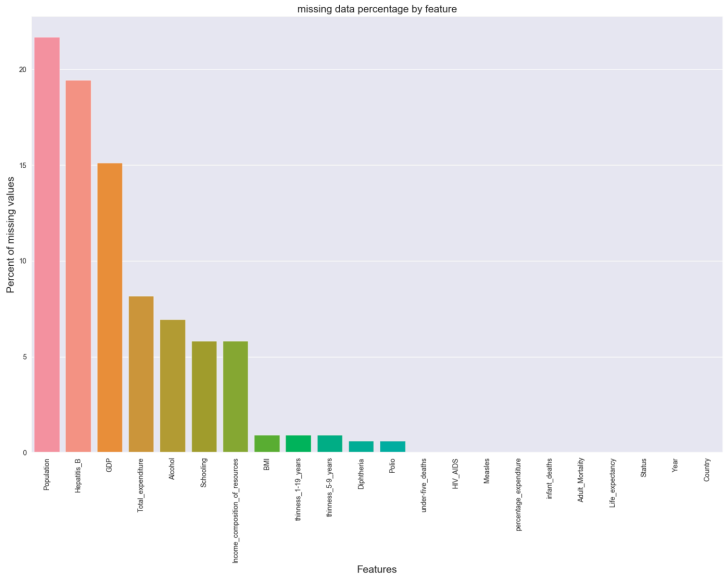
(a) ”Country”: We apply one-hot encoding on the feature.

(b) “Status”: There are two kinds of situations in the features, developing and developed. Because the developed countries can have better live quality and health care environment, it can have more contribution to output “Life Expectancy” than developing ones. Therefore, we assigned “2” to developed countries and “1” to developing countries.

After transforming, we would like to combine “status” with other numerical features.

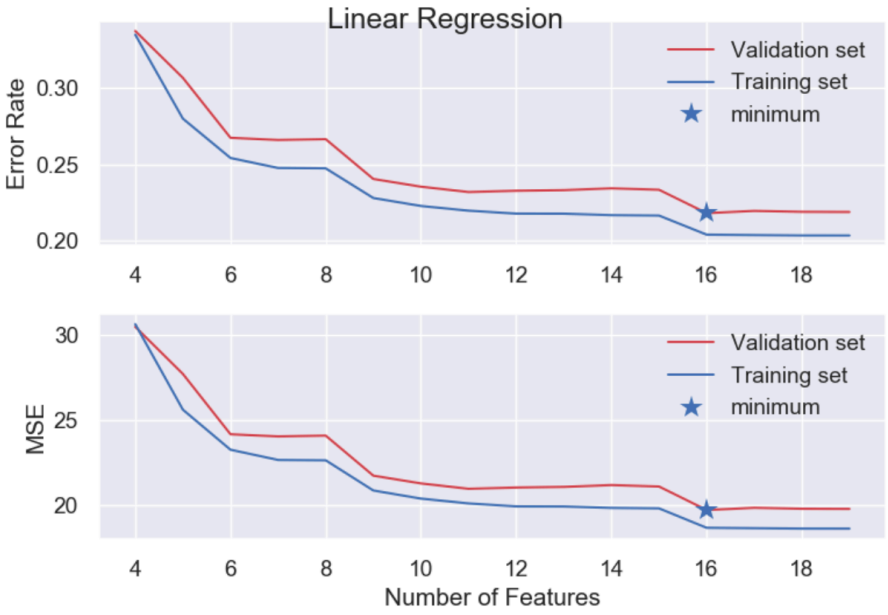
3.2.3. Missing values in features

We can observe the percentage of missing data in each feature first. If there are not a lot of missing values, we can delete the data points with missing values directly. However, if the dataset contains a lot of missing values, we need to fill them with some values. Therefore, we can take a look at the missing proportion in each feature first.

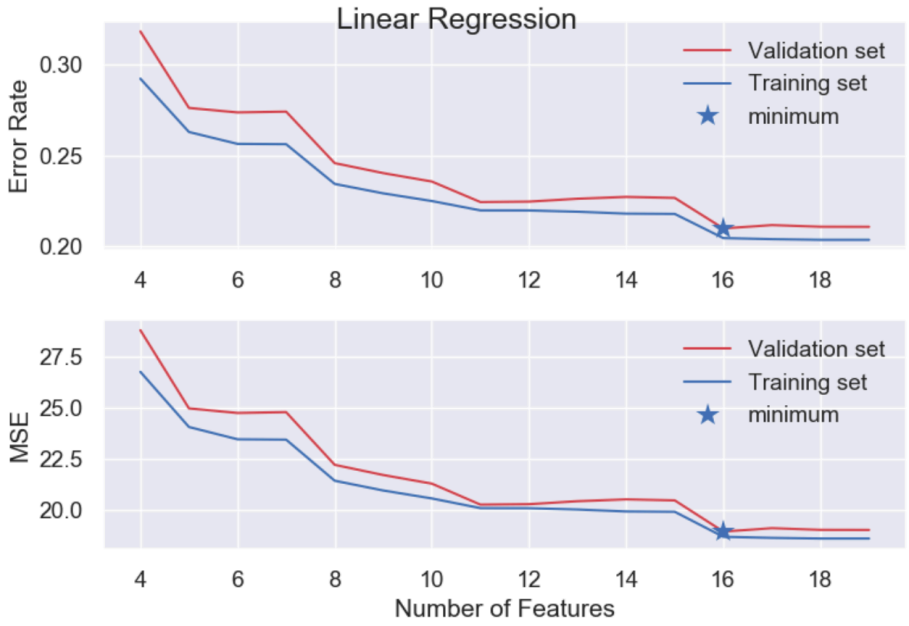


Because we have limited data and the missing percentage is higher that 20% in feature “Population”, we need to find the best way to fill the missing value. We compare different techniques in Lin with maximum feature=3 in each draw and found that we can reach the lowest MSE in validation set by filling the missing value with “mean”.

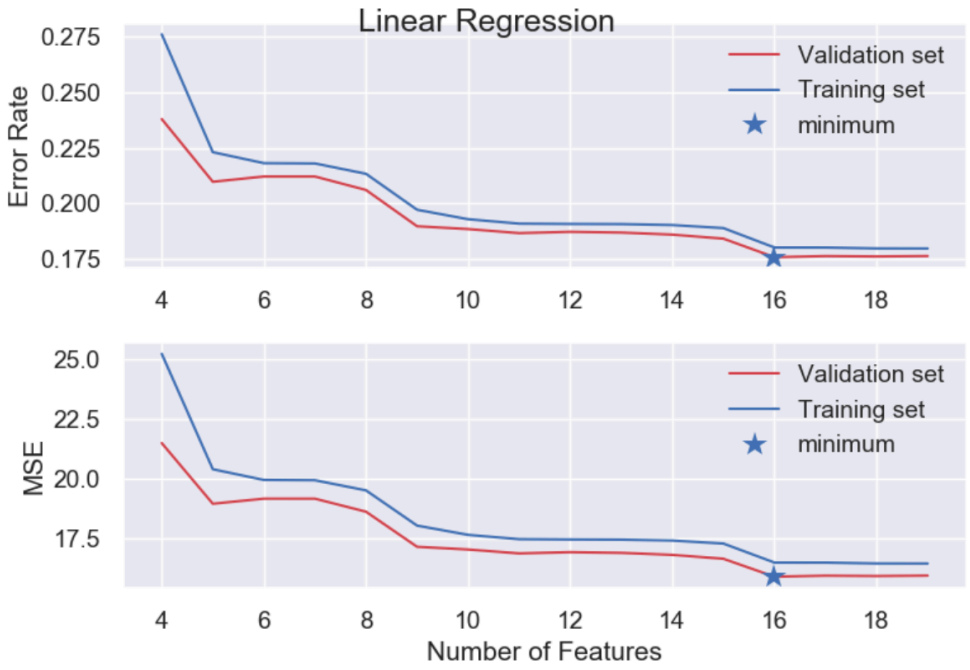
(a) Fill the missing values with “0”.
MSE=19.7050, Error rate=0.2182 (when there are 16 features)



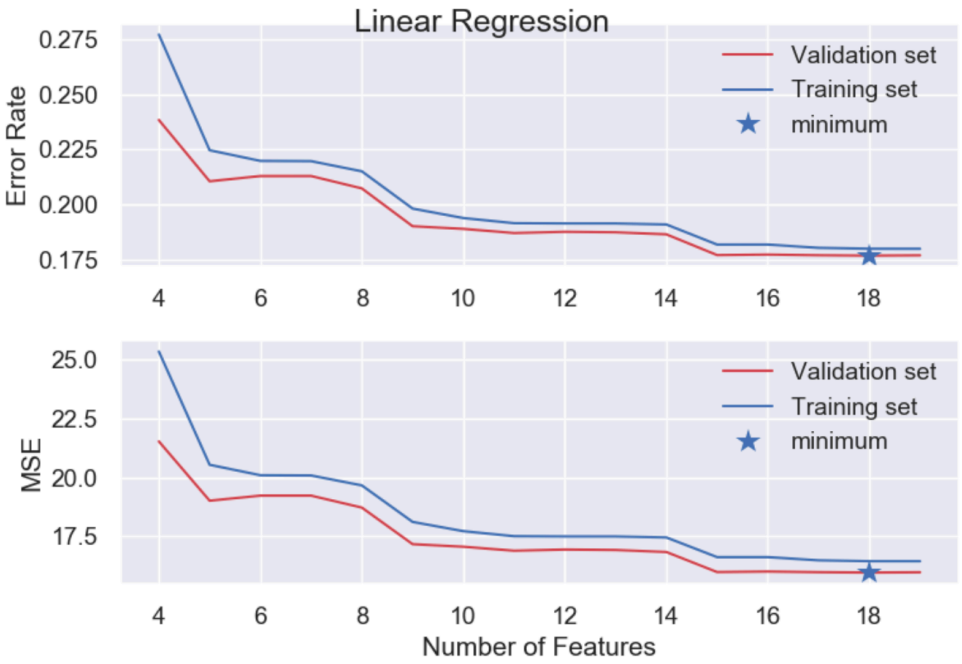
(b) Fill the missing values with "-1".
MSE=18.9485, Error rate=0.2098 (when there are 16 features)



(c) Fill the missing values with "mean".
MSE=15.8867, Error rate=0.1759 (when there are 16 features)



(d) Fill the missing values with "median".
MSE=15.9664, Error rate=0.1768 (when there are 18 features)



From above, we can get the table:

Methods	Linear Regression	
	Error rate	MSE
Fill with "0"	0.2182	19.7050
Fill with "-1"	0.2098	18.9485
Fill with "mean"	0.1759	15.8867
Fill with "median"	0.1768	15.9664

3.2.4. Outliers

We compared the results to deal with the outliers or not and we found that the MSE value will be lower if we address it.

(a) Handle with outliers: (MSE=12.9198, Error rate=0.1431)

We calculated the inter-quartile range (IQR) and used its multiples to define the outliers. The IQR = Q3 - Q1, where Q1 is the first quartile, and Q3 the third quartile.

The potential outliers lie outside the range of :

$$[Q1 - (1.5 \times IQR), Q3 + (1.5 \times IQR)]$$

When outliers are higher than the range, we will replace the values with the maximum value within the range; if outliers are smaller than the range, we will replace them with the minimum value within the range.

(b) Not to handle it: (MSE=15.8867, Error rate=0.1759)

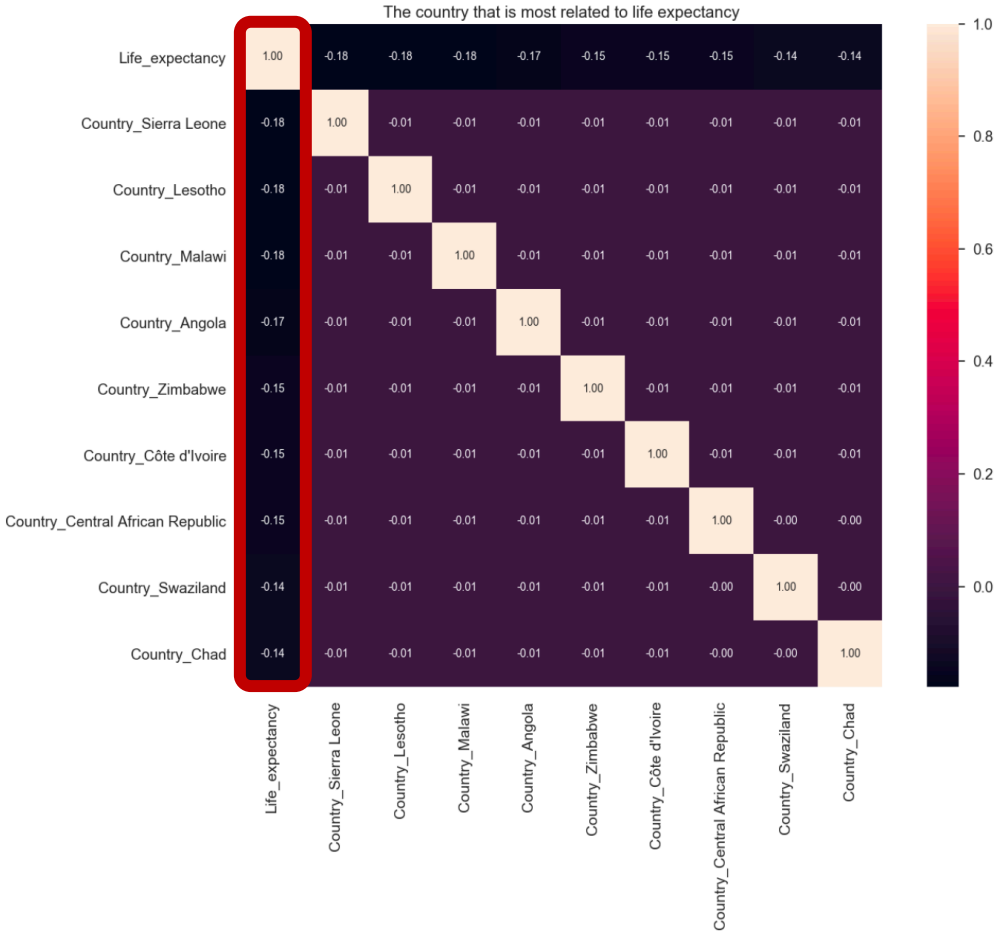
Methods	Linear Regression	
	Error rate	MSE
Handle with outliers	0.1431	12.9198
Not to handle outliers	0.1759	15.8867

3.2.5. Dimensionality Reduction

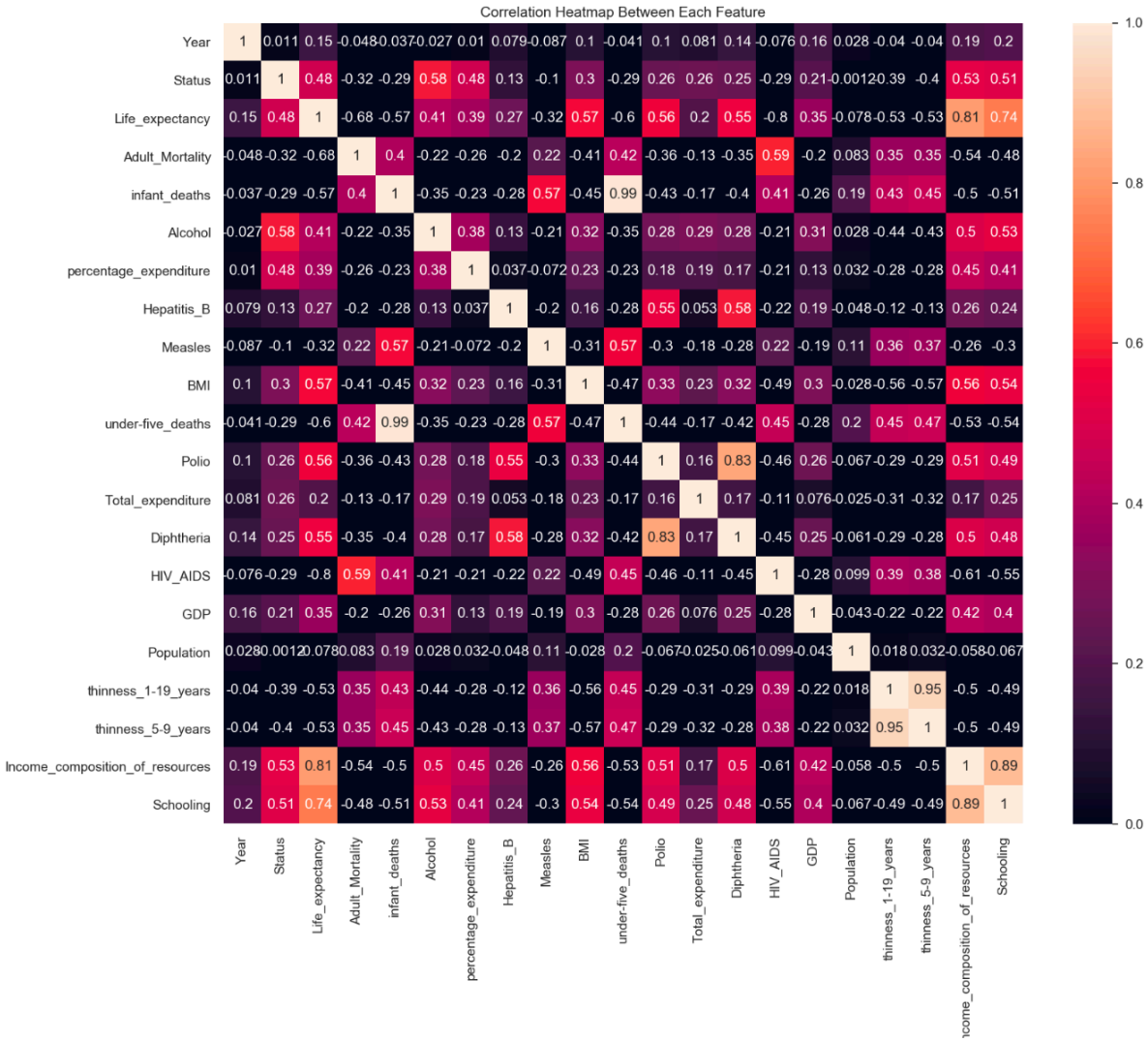
(a) method 1:

As we mentioned in section 3.2.2 Categorical feature, we have a one-hot encoding result and other numerical features now.

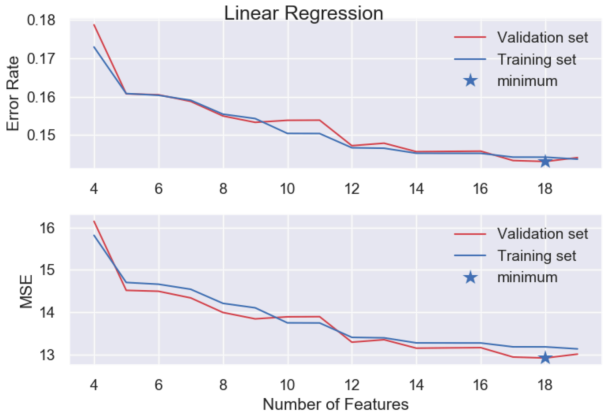
We first have a look at the correlation between the one-hot encoding result and “Life Expectancy”. Since there are 183 countries in total, we only pick up the top 9 which are the most related to our target. From the heatmap shown below, we can find that the highest correlation after taking absolute value is 0.18.



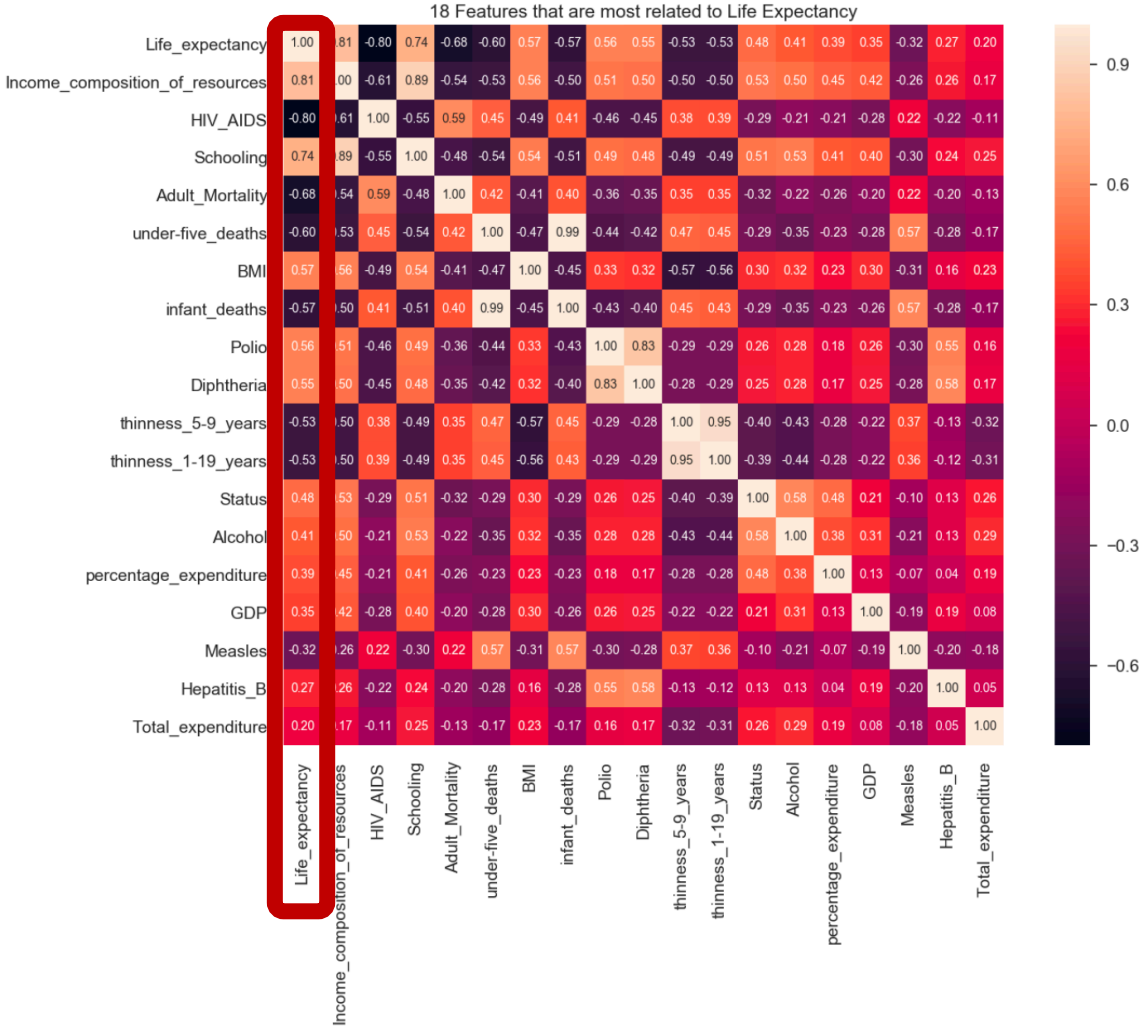
Second, we observe the correlation between other numerical features and “Life Expectancy” as we can see in the following figure.



In order to reduce the dimension, we first calculate MSE in validation set in linear regression when we pick up different numbers of features which are higher than other features in correlation after taking absolute value.



As we can see in the figure above, we can obtain the minimum MSE when we pick up 18 features which are the most correlated to the target to train the linear regression model.



As we can see in the figure shown above, the lowest correlation after taking absolute value is 0.2 which is much higher than the highest value in the one-hot encoding result 0.18, therefore, we can ignore that result directly. Now, we successfully reduce the dimension to 18 features.
In the following procedure, we will use the dimension to do some of the model and parameter selections.

(b) Method 2:
We use Random Forest Regression to see the importance of each model. The importance of each feature in the model is shown as follows.

Feature Name	Importance
HIV_AIDS	0.549662
Income_composition_of_resources	0.192459
Adult_Mortality	0.112182

Schooling	0.026946
BMI	0.021404
under-five_deaths	0.019708
Year	0.011290
Total_expenditure	0.008037
infant_deaths	0.007660
percentage_expenditure	0.006656
Polio	0.005255
thinness_5-9_years	0.005186
thinness_1-19_years	0.005146
Measles	0.004821
Population	0. 004214
Diphtheria	0. 003884
GDP	0.003253
Hepatitis_B	0.002693
Status	0.000100
Country	ignore

We remove the feature in the Random Forest Regression one by one to find the optimal dimensionality. We will state the process in section 3.4.3.

3.2.6. Standardization

After finishing the steps mentioned above, we calculate the mean and standard deviation in training (Original/ New) data (see section 3.3) to standardize each feature in training (Original/ New) data with 0 mean and unit variance. Then, we apply standardization with the mean and standard deviation we get to transform test/ validation data individually.

3.3. Dataset Methodology

The dataset contains 2938 sample points in total and we remove 10 points that have no value on our target feature at first. In the training process, we try both validation set and cross validation methods to calculate MSE and find the optimal parameter and model.

3.3.1 Validation set

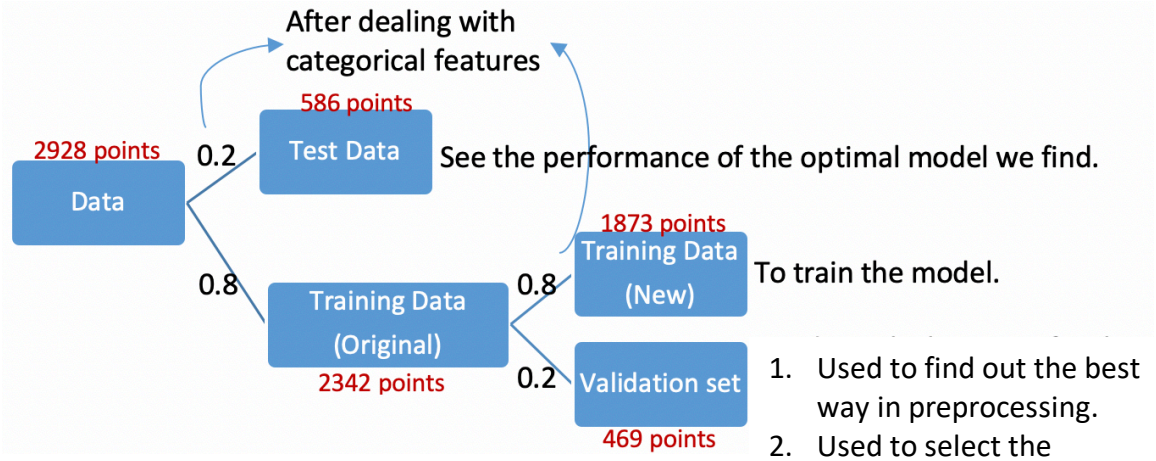
We separate the data after dealing with categorical features and the process are shown in the following figure.

We use validation set to calculate MSE in:

(a) Preprocessing: We use validation set to pick up the method with the lowest MSE as the best way to fill the missing value in inputs (section 3.2.3.). In addition, we also use the dataset to reduce the dimensionality as we mentioned in section 3.2.5..

(b) Parameter and model selections: We apply validation set to find the optimal parameter and model in Random Forest Regression, Linear

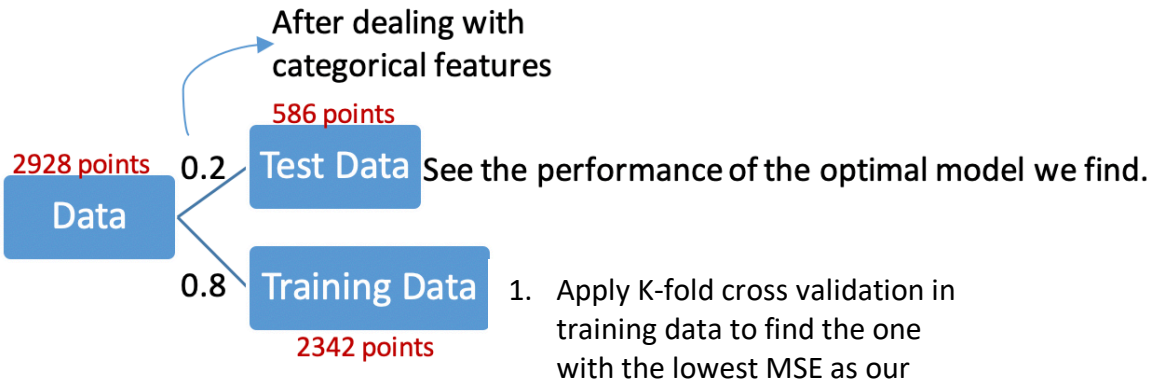
Regression, Ridge Regression and Lasso Regression as we stated in section 3.4. Training Process.



1. Used to find out the best way in preprocessing.
2. Used to select the optimal parameter and model from the model we trained.

3.3.2 Cross Validation

We use the training data (2342 points) to apply cross validation, calculate mean MSE and find the optimal k-fold and model in Ridge Regression and Lasso Regression. In addition, after obtaining the final model, we will use the whole training data to train the model again.



1. Apply K-fold cross validation in training data to find the one with the lowest MSE as our optimal model.
2. After obtaining the optimal parameter and model, use the whole training set to train the model again.

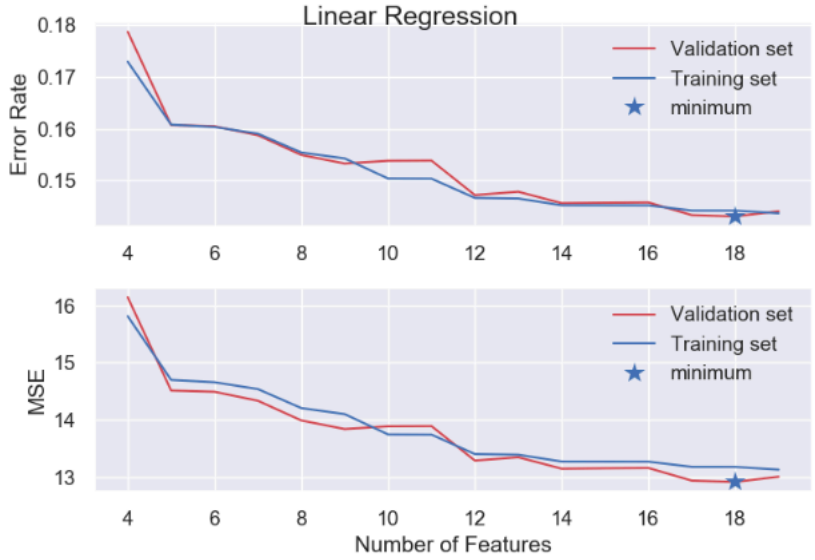
3.3.3 Test data

There are 586 points in test dataset. After obtaining the final model from different regression methods, we will use the dataset only once to see the performance of the model we obtain.

3.4. Training Process

3.4.1. Linear Regression

We would like to try Linear Regression model first because it is the simplest model. We use new training data to train our model with different numbers of features we pick up according to its correlation with feature "Life expectancy". Then, use validation set to calculate MSE for each of them and find the optimal parameter.



We get the minimum MSE=12.9198 and Error rate=0.1431 when there are 18 features. (The dimension will be 18 features in other models we try.)

3.4.2. Ridge Regression

The model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm, therefore, it sometimes does better than linear regression.

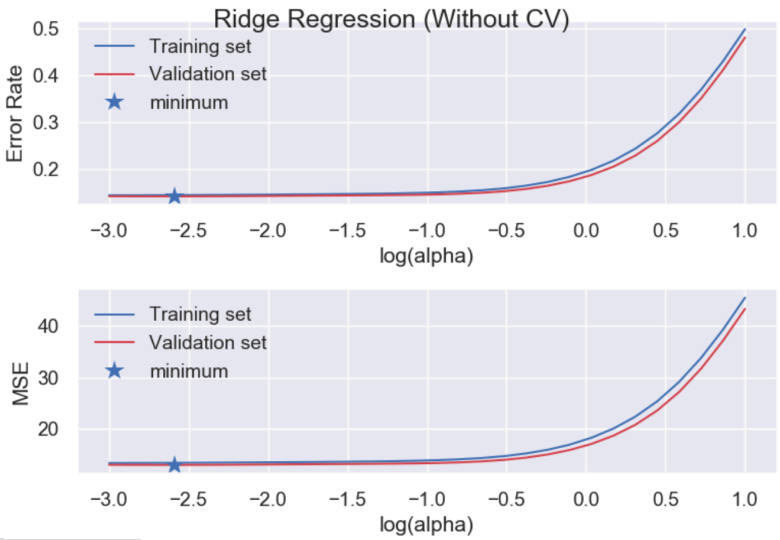
$$W^{*}=\underset{w}{\operatorname{argmin}} \|y - Xw\|_2^2 + \alpha \|w\|_2^2$$

We pick up α in the range $10^{-3}$to 10^{1} for 30 values evenly separated.

Validation

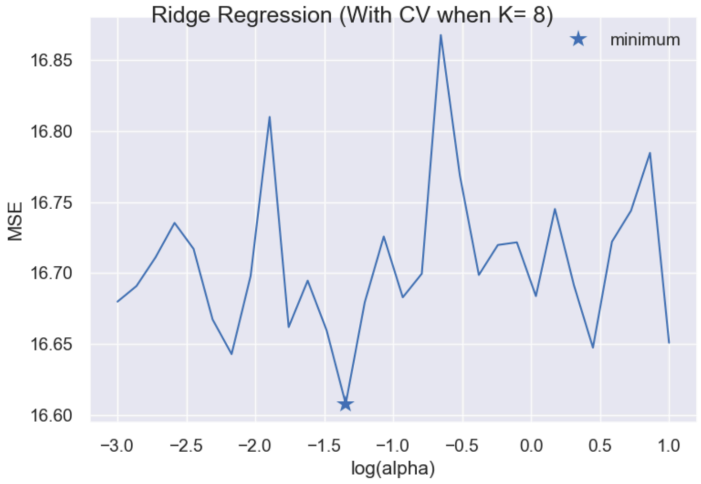
We use new training data to train the model for different α, then, use validation data to calculate MSE and find the best α.

We get the minimum MSE= 12.8264 and error rate= 0.1423 when α = 0.002593.



Cross Validation

We use original training data to do K-fold cross validation with different α (K is an integer in the range from 5 to 10). After training the model, we can compare the average MSE for each (K, α) pair and pick up the best one. We get the minimum MSE=16.608 when the best pair (K, α) = (8, 0.04520). The following plot shows the error rate and MSE for different α in 8-fold cross validation.



3.4.3. Lasso Regression

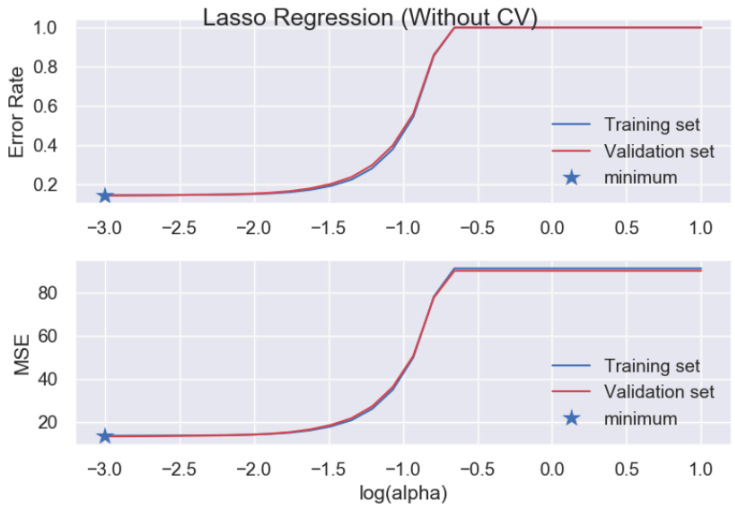
Lasso Regression also adds a penalty term to the loss function as well as Ridge Regression. The only difference between them is instead of taking the square of the coefficients, Lasso Regression takes magnitudes into account. This type of regularization (L1) can lead to zero coefficients i.e. some of the features are completely neglected for the evaluation of output. So Lasso regression not only helps in reducing over-fitting but it can help us in feature selection.

$$W^{*}=\underset{w}{\operatorname{argmin}} \|y - Xw\|_2^2 + \alpha \|w\|$$

We pick up α within the range $10^{-3}$to 10^{1} for 30 values evenly separated.

Validation

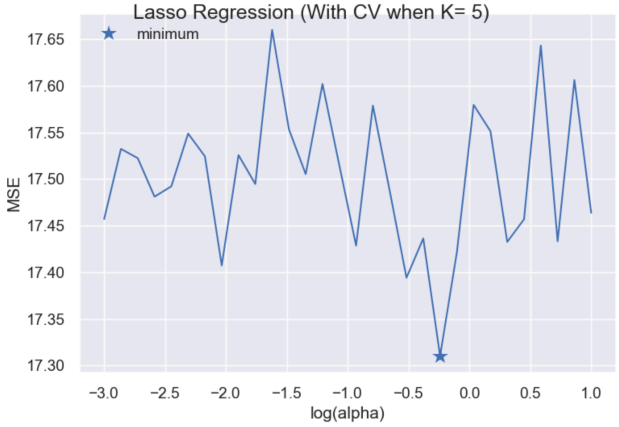
We use new training data to train the model for different α, then, use validation data to calculate MSE and find the best α.



We get the minimum MSE= 13.1457 and Error rate= 0.1456 when α= 0.001.

Cross Validation

We use training data (original) to do K-fold cross validation with different α (K is an integer in the range from 5 to 10). After training the model, we can compare the average MSE for each (K, α) pair and pick up the best one. We get the minimum MSE= 17.310 and when the best pair (K, α) = (5, 0.5736). The following plot shows the error rate and MSE for different α in 5-fold cross validation.



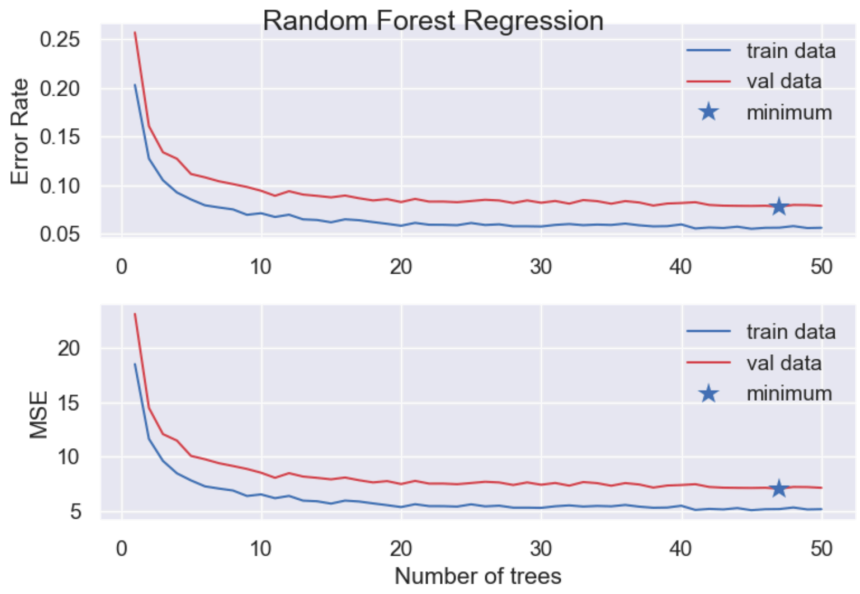
3.4.3. Random Forest Regression

Since Random Forest Regression is one of the most effective methods, we would like to implement the algorithm to the dataset. We use the new training data to train the model. At each iteration, first create a smaller training set, bag, randomly drawn from the new training data (1/3 of the training set size).

The fixed parameters:
Bootstrap = True
Max number of features in each draw= 3
Number of trees = 1 to 50.
We repeat the experiment 10 times for each value of number of trees (selecting different bag samples every time).

Method 1: Fixed the number of features

After training the model, we use validation set to calculate mean MSE and obtain the optimal number of trees. We reach the minimum MSE=7.063 and error rate=0.078 when there are 47 trees



Method 2 :Remove the feature one by one according to its importance

When we train our model in each iteration, we calculate and remove the least importance feature in the dataset, then, find the optimal parameter and model which are shown in the table.

#drop features	Feature Name	MSE	Optimal nTrees
1	Country	6.456	38
2	Status	6.389	46
3	Hepatitis_B	6.374	48
4	Infant_deaths	6.413	46
5	GDP	6.422	41
6	Measles	6.369	40
7	Population	6.294	44
8	Percentage_expenditure	6.313	43
9	Diphtheria	6.087	38
10	Polio	6.366	38
11	Thinness 5-9 years	6.293	41

12	Total_expenditure	6.203	39
13	Thinness 1-19 years	6.256	39
14	Alcohol	6.352	49
15	Year	6.216	29

From the table above, we could get the optimal MSE=6.087 when we remove 9 features.

3.5. Model Selection and Comparison of Results

We would like to select the final model from the optimal models that we get from different algorithms as we stated in section 3.4.

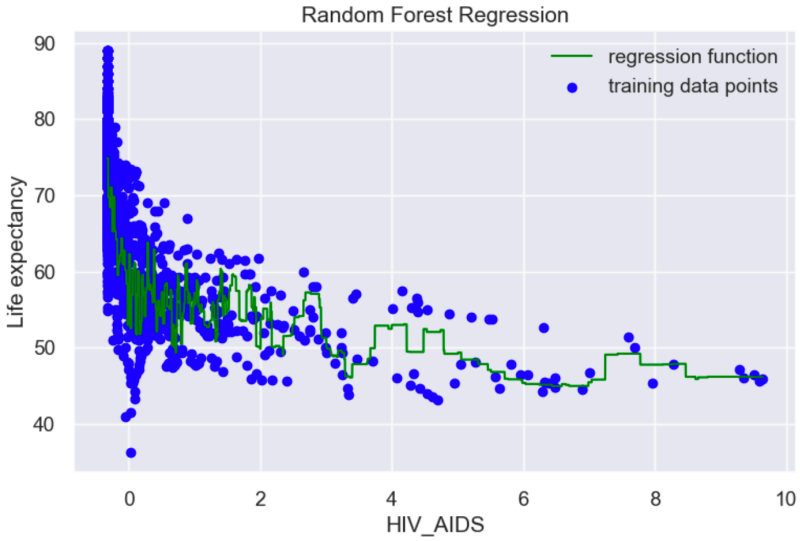
Methods		MSE		Parameters
Linear Regression	Validation	12.9198		N_features=18
Ridge Regression	Validation	12.8264		α = 0.002593
	Cross Validation	16.608		(K, α) (5, 0.5736)
Lasso Regression	Validation	13.1457		α =0.001
	Cross Validation	17.310		(K, α)=(5,0.5736)
Random Forest Regression	Validation	Method1	7.063	nTrees=47
		Method2	6.087	nTrees=38

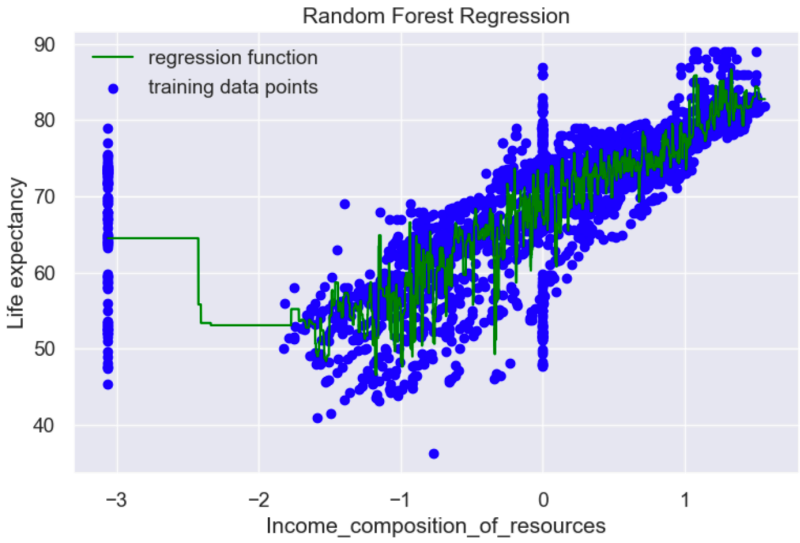
At first, I thought linear regression might not be better than ridge regression or lasso regression. The reason causes the result might be that we based on linear regression to find the optimal number of features, however, it might not be the best number of features for the other algorithms.
After comparing with all of the methods we tried, we obtained the optimal model "Random Forest Regression" with 38 trees.

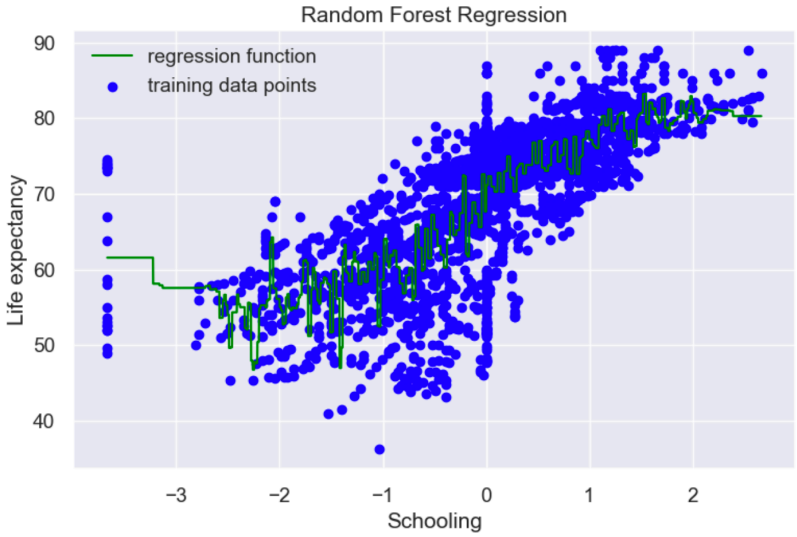
In our final model "Random Forest Regression" with nTrees=38 and 9 dropped features, we analyze the importance of the left features. The top 4 features in importance are shown in the following table.

Feature	Importance
HIV_AIDS	0.581111
Income_composition_of_resources	0.150113
Adult_Mortality	0.132056
Schooling	0.062197

The 2D plots of these important features after standardization and the final regression function are shown as following figures.







The code for the plot above was modified from https://www.geeksforgeeks.org/random-forest-regression-in-python/ , Random Forest Regression in Python.[4]

4. Final Results and Interpretation

From all of the steps mentioned above, we obtain our final model which is in Random Forest Regression with 9 features removed, which are Country, Status, Hepatitis_B, Infant_deaths, GDP, Measles, Population, Percentage_expenditure and Diphtheria.
In each iteration, we create a smaller training set, bag, randomly drawn from the original training data (1/3 of the training set size) and bootstrap = True.
We repeat the experiment 10 times for each value of number of trees (selecting different bag samples every time) to calculate its mean MSE and error. After setting all the parameters mentioned above, we use the whole training set to train the model. Second, use the test dataset to see the performance of our final model.

Test dataset	Baseline	The final model
MSE	88.56	5.474 with σ=0.367
Error rate (1 - r2 score)	100%	6.18% with σ = 0.0045

From the tables and plots given above, we can know that:

1. The features influence the target the most are ”HIV_AIDS”, ”Adult Mortality”, ” Income composition of resources” and “schooling”.
2. The feature” HIV_AIDS” is deaths per 1000 live births caused by HIV/AIDS (0-4 years) and it makes sense that the lower the deaths of the baby because of HIV/AIDS, the better the health care the country is, therefore, the higher the life expectancy is and vice versa.
3. For most countries, the Income composition of resources is proportional to life expectancy.

4. The Adult Mortality Rates measure the probability of dying between 15 and 60 years per 1000 population). It makes sense since the lower the Adult Mortality is, the lower the life expectancy is.
5. The higher the education people have in a country, the better developed the country is, therefore, the higher the life expectancy is.
6. We might think more information gives better results. However, as the table in section 3.4.3 to remove the feature one by one, the fewer the features sometimes bring the lower MSE.

Time Complexity [6]

The complexity of random forest can be approximated as following.

Training	Prediction
O($\boldsymbol{n^2pn_{trees}}$)	O(pn_{trees})

n is the number of training sample, p the number of features, $n_{trees}$is the number of trees.

For our final model:

Theoretical Point of view

Training complexity: O($\boldsymbol{n^2pn_{trees}}$)= O($\mathbf{2342^2x12x38}$)

Prediction complexity: O($\boldsymbol{pn_{trees}}$)= O($\mathbf{12x38}$)

Practical Point of View (focus on sklearn implementations**)**

The assumptions will be that the complexities take the form of $O(n^{\alpha}p^{\beta})$ and α and β will be estimated using randomly generated samples with n and p varying. Then, using a log-log regression, the complexities are estimated. （In random forest, α=1.21 and β=0.89）

Training complexity: O($n^{\alpha}p^{\beta}$)= O($\mathbf{2342^{1.21}x38^{0.89}}$)

Prediction complexity: O($n^{\alpha}p^{\beta}$)= O($\mathbf{586^{1.21}x38^{0.89}}$)

Though this assumption is wrong, it should help to have a better idea of how the algorithms work and it will reveal some implementation details/ difference between the default settings of the same algorithm that one may overlook.

Out of Sample Error

(a) Calculate from sklearn [5] :

In random forest regression, we create a smaller training set, bag, randomly drawn from the original training data (1/3 of the training set size). We can take the bag as our training set and use the training data as validation set. For each tree, we have different validation sets. We would average all the trees where that rows is not used for training to calculate our prediction. If we have many trees, it is likely that all of the rows appear for several times in these out-of-bag samples. Then, we can calculate R2 on these out-of-bag(=out-of-sample) predictions.

For our model, $E_{out}(hg)$= 1 - out-of-bag_score=0.0662.

(b)Calculate the upper bound for $E_{out}(hg)$:

Assume the tolerance $\delta=0.1$ and use R2 to calculate the error. (M=1)

$$E_{out}(hg) \leq E_{test}(hg) + \sqrt{\frac{1}{2N} ln \frac{2M}{\delta}}$$

$$\Rightarrow \quad E_{out}(hg) \leq 0.0618 + \sqrt{\frac{1}{2x586} ln \frac{2x1}{0.1}}$$

$$\Rightarrow \quad E_{out}(hg) \leq 0.0618 + 0.05 = 0.1118 \quad \text{with probability} \geq 1 - \delta$$

From above, $E_{out}(hg)=0.0662$ in (a) is located within the range that

$$E_{out}(hg) \leq 0.1118$$

5. Summary and conclusions

From the analysis of life expectancy in different countries, we find that it is influenced by multiple factor. In general, we might think that Life Expectancy has more correlation with eating habits, lifestyle, exercise, smoking, drinking alcohol. However, we find that the feature that influences life expectancy the most is “HIV/AIDS” which is deaths per 1000 live births caused by HIV/AIDS (0-4 years).

To make a conclusion, after comparing so many models, random forest has the best performance in predicting life expectancy and it successfully improves the MSE from the baseline=88.56 to 5.608 with σ=0.36. In order to improve, we can also try different parameters, like max depth, max features, minimum number of samples required to split an internal node during the process next time.

6. References

[1] SHUBHAM JAIN, "A comprehensive beginners guide for Linear, Ridge and Lasso Regression in Python and R.", JUNE 22, 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/.

[2] "What are outliers in the data". [Online]. Available: https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/.

[3] Mario Lewis,"The Importance of Understanding Outliers.", October 28 2016. [Online]. Available: https://www.mytechlogy.com/IT-blogs/13804/the-importance-of-understanding-outliers/#.Xe2YvC2_snV

[4] "Random Forest Regression in Python." [Online]. Available: https://www.geeksforgeeks.org/random-forest-regression-in-python/

[5] Navnina Bhatia,"What is Out of Bag (OOB) score in Random Forest?", Jun 26,2019. [Online]. Available: https://towardsdatascience.com/what-is-out-of-bag-oob-score-in-random-forest-a7fa23d710

[6] "Computational Complexity of Machine Learning Algorithms", April 16,2018. [Online].Available:https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/

[7] Julia Kho," Why Random Forest is My Favorite Machine Learning Model", Oct 19, 2018. [Online]. Available: https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706