Improving Your Life Expectancy

Chong Wei Kang U2121461B
Joel Tan U2122416C
Sua Qi Rong U2122411D

TABLE OF CONTENTS

01

Introduction

Problem Definition , Motivation & Dataset Used

02

Data Preparation & EDA

Cleaning of data and initial data-driven insights

03

Machine Learning

Multi-variate Regression with SKLearn, Random Forest and TensorFlow

04

Conclusion

Outcome & Data Driven-Insights

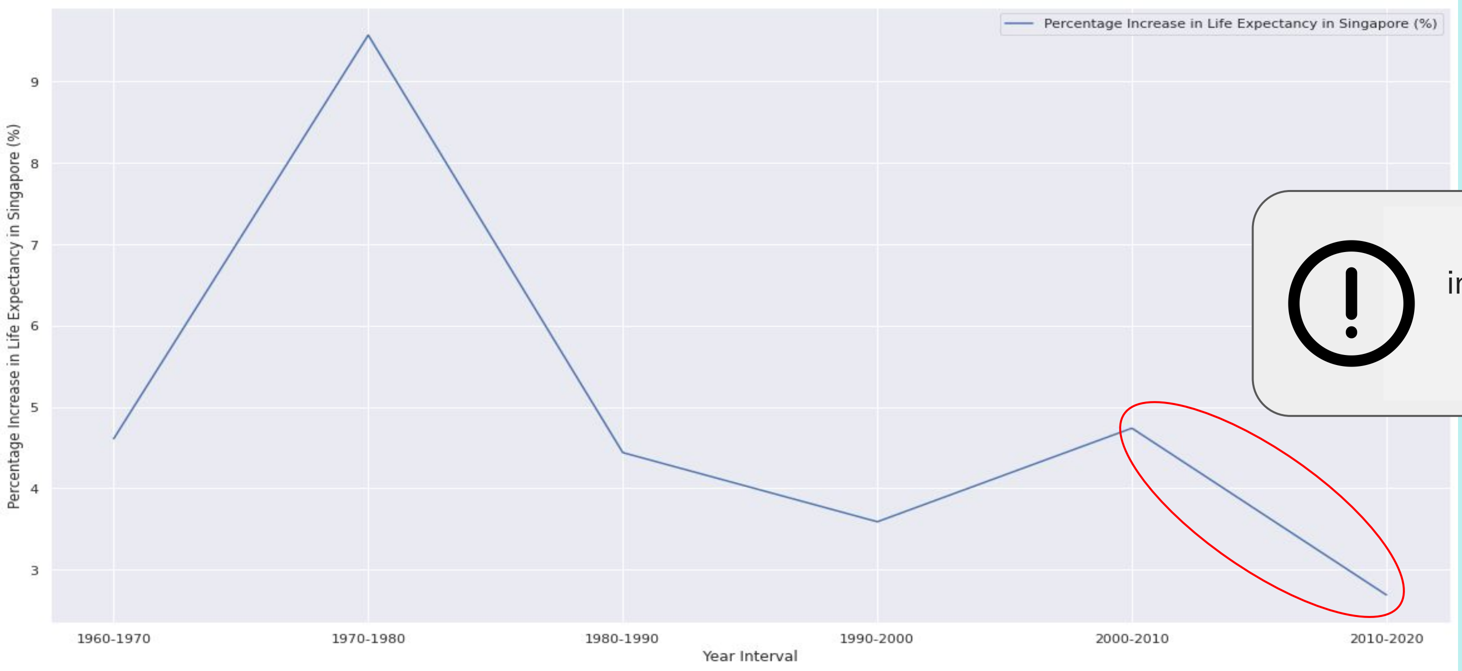
01. INTRODUCTION

“Data is the new oil.” — Clive Humby

Problem Definition

What should we do to **effectively** increase the life expectancy of Singapore's population in today's context? What are some **main areas** of concern to prioritise and tackle?

Motivation



Rate at which LE is increasing in the past decade has **slowed significantly.**

Why is this so? What are some things that Singapore should zoom in and **focus on** in order to increase Life Expectancy by a larger rate again?

Motivation

2022 Life Expectancy

1	Hong Kong	85.29
2	Japan	85.03
3	Macau	84.68
4	Switzerland	84.25
5	**Singapore**	**84.07**

2025 Life Expectancy

1	**Singapore**	**88.00**
2	Hong Kong	87.00
3	Japan	86.00
4	Macau	85.00
5	Switzerland	84.50

Let's make Singapore #1!

Life Expectancy Dataset

Variables

- Life Expectancy
- Alcohol
- Percentage Expenditure
- .... **many more!**

Data Points

- 2938 Rows
- 22 Columns
- Collected from WHO & UN website

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...
...	...	...	...	...	...	...	...	...	...	...	...
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.000000	68.0	31	...
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.000000	7.0	998	...
2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.000000	73.0	304	...
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.000000	76.0	529	...
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.000000	79.0	1483	...

2938 rows × 22 columns

02a. Data Preparation

"Without a systematic way to start and keep data clean, bad data will happen." — Donato Diorio

Data Preparation

Dropping Data

Dropping 'Year' and 'Country'

```
#Drop Country and Year
life_expectancy_data = life_expectancy_data.drop(columns = ['Country','Year'])
```

```
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 20 columns):
 #   Column
---  ------
 0   Status
 1   Life expectancy
 2   Adult Mortality
 3   infant deaths
 4   Alcohol
 5   percentage expenditure
 6   Hepatitis B
 7   Measles
 8    BMI
 9   under-five deaths
 10  Polio
 11  Total expenditure
 12  Diphtheria
 13   HIV/AIDS
 14  GDP
 15  Population
 16   thinness  1-19 years
 17   thinness 5-9 years
 18  Income composition of resources
 19  Schooling
```

Data Preparation

Correcting Variable Names

Wrongly written names and weird spaces must be taken care of

```
 #   Column                           Non-Null Count   Dtype
---  ------                           --------------   -----
 0   Status                           2938 non-null    object
 1   Life expectancy                  2928 non-null    float64
 2   Adult Mortality                  2928 non-null    float64
 3   infant deaths                    2938 non-null    int64
 4   Alcohol                          2744 non-null    float64
 5   percentage expenditure           2938 non-null    float64
 6   Hepatitis B                      2385 non-null    float64
 7   Measles                          2938 non-null    int64
 8    BMI                             2904 non-null    float64
 9   under-five deaths                2938 non-null    int64
 10  Polio                            2919 non-null    float64
 11  Total expenditure                2712 non-null    float64
 12  Diphtheria                       2919 non-null    float64
 13   HIV/AIDS                        2938 non-null    float64
 14  GDP                              2490 non-null    float64
 15  Population                       2286 non-null    float64
 16   thinness  1-19 years            2904 non-null    float64
 17   thinness 5-9 years              2904 non-null    float64
 18  Income composition of resources  2771 non-null    float64
 19  Schooling                        2775 non-null    float64
```

thinness **10**-19 years

Data Preparation

Addressing NA Values

Filling NA values with the median of the original data

```
life_expectancy_data.isnull().sum()
```

Variable	No of NA
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
HIV/AIDS	0
GDP	448
Population	652
thinness 10-19 years	34
thinness 5-9 years	34
Income composition of resources	167
Schooling	163

Data Preparation

Removing Outliers

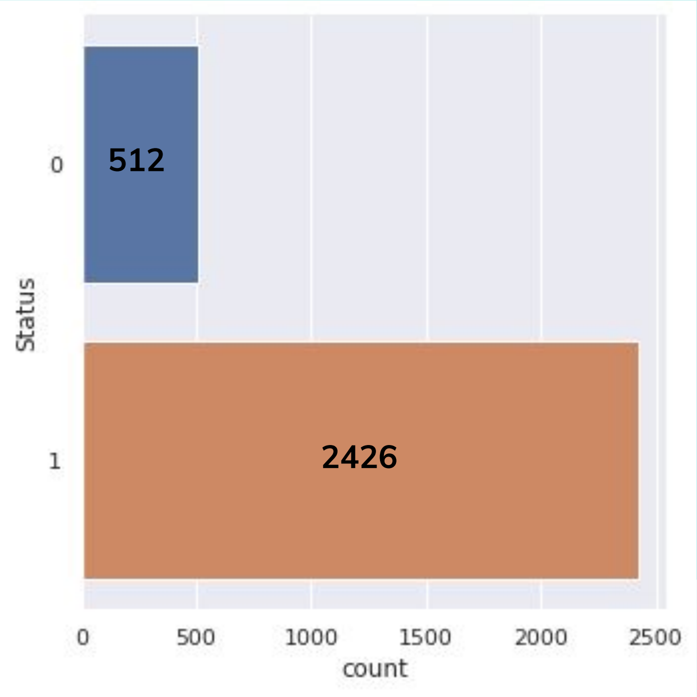
Removing outliers which are +- 1.5 IQR from Q1 and Q3

```
outliers_variables = ['Adult Mortality','Alcohol','Schooling','Income composition of resources',
                      'thinness 10-19 years','thinness 5-9 years','Life expectancy']
all_outliers_indices = []
sum = 0
for var in filtered_data:
    if var in outliers_variables:
        Q1 = filtered_data[var].quantile(0.25)
        Q3 = filtered_data[var].quantile(0.75)
        IQR = Q3-Q1

        #create new column to identify outliers
        filtered_data['Outlier'] = ((filtered_data[var] < (Q1 - 1.5 * IQR)) | (filtered_data[var] > (Q3 + 1.5 * IQR)))
        #sum of outliers
        no_of_outliers = filtered_data['Outlier'].sum()
        sum += no_of_outliers

        #This is just a check against the number of outliers found above to ensure consistency.
        print(f'Column {var} has {no_of_outliers} outliers.')

        outlierindices = filtered_data.index[filtered_data['Outlier'] == True]
        # print(outlierindices)
        for index in outlierindices:
            if index not in all_outliers_indices:
                all_outliers_indices.append(index)
```

```
# Removing all rows with the outliers
filtered_data.drop(axis = 0, index = all_outliers_indices, inplace = True)
filtered_data
```

Data Preparation

Label Encoder	
Status (Developed)	Status (Developing)
0	1

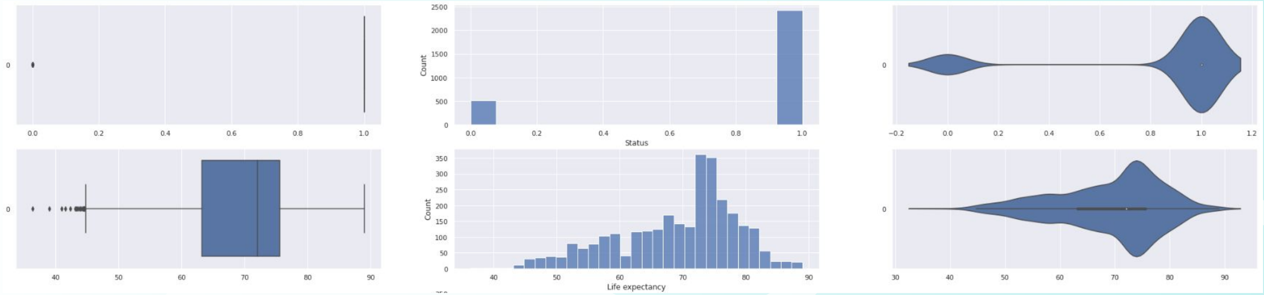


02b. Exploratory Data Analysis

"Torture the data, and it will confess to anything."
— Ronald Coase

Exploratory Data Analysis

```python
count = 0
for var in life_expectancy_data:
    sb.boxplot(data = life_expectancy_data[var], orient = "h", ax = axes[count,0])
    sb.histplot(data = life_expectancy_data[var], ax = axes[count,1])
    sb.violinplot(data = life_expectancy_data[var], orient = "h", ax = axes[count,2])
    count += 1
```



Plotting the distribution of all variables to observe any patterns (Boxplot, Histogram & Violinplot)

Initial Data-Driven Insights

Insight 1

Mean Life Expectancy - 69.22

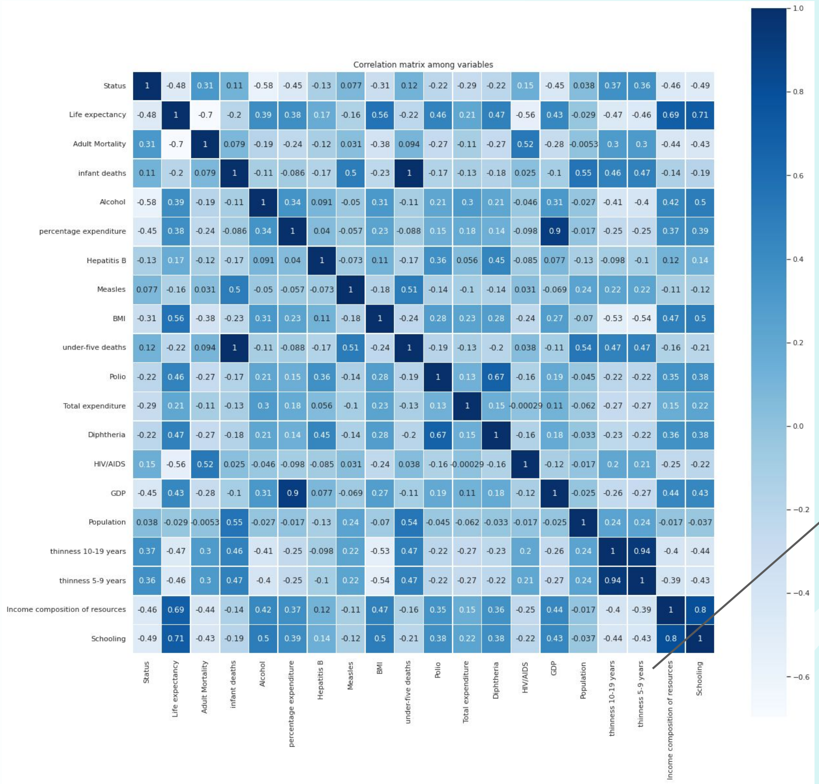
There is much room for improvement

Insight 2

Distributions for each individual variable do not suggest much

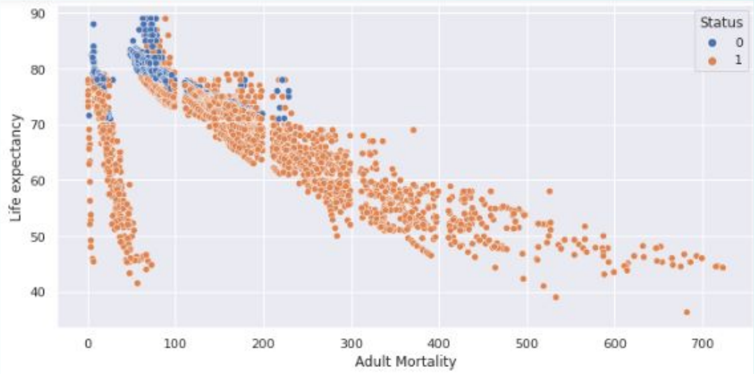
Find more detailed insights of how each variable impacts Life Expectancy instead.

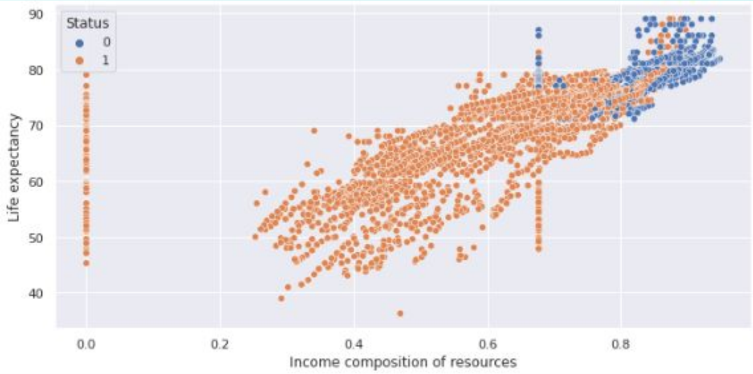
Correlation Heatmap

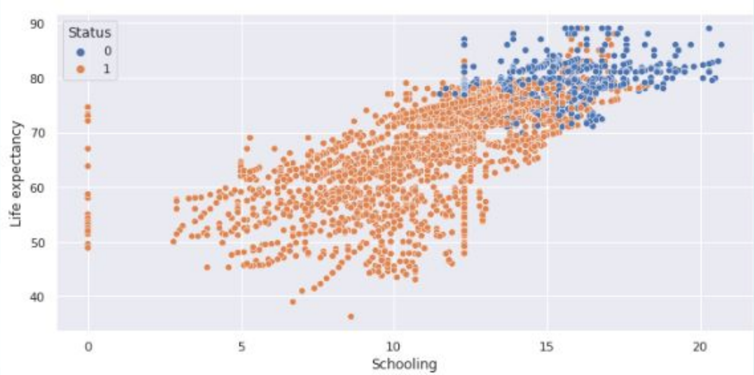


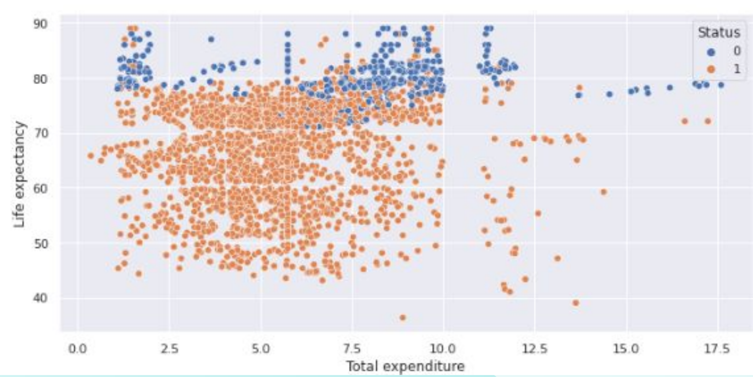
Some notable high & lows	
Schooling 0.71	Population -0.029
Internal composition of resources 0.69	Hepatitis B 0.17
BMI 0.56	Measles -0.16
Adult Mortality -0.7	Total Expenditure 0.21

Exploratory Data Analysis

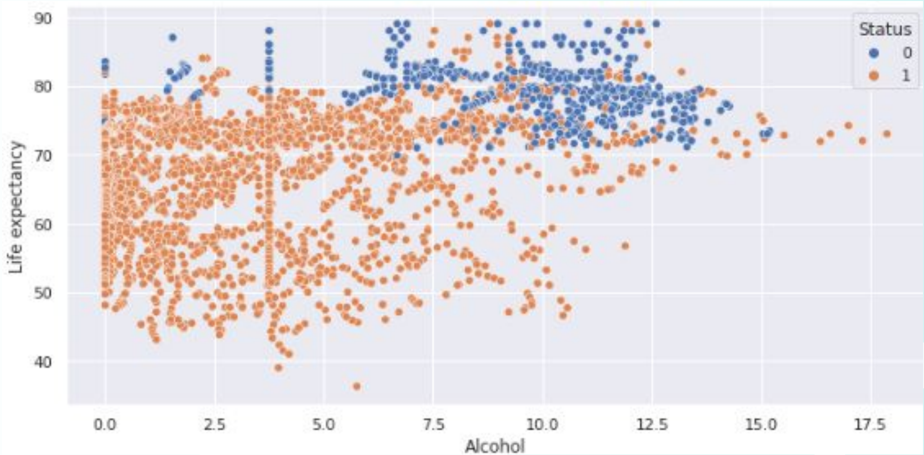








Initial Data Driven Insights



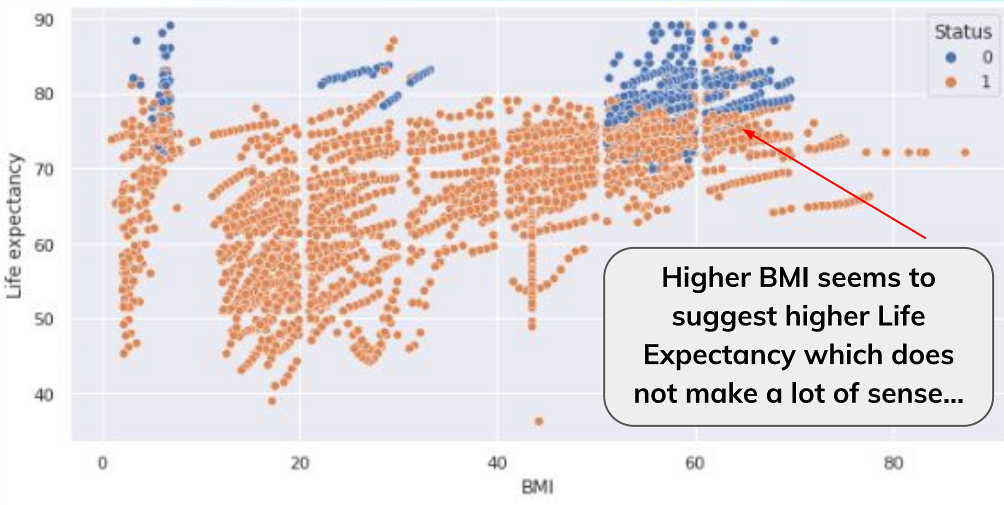
Insight 1

No clear correlation between alcohol and LE

Developed countries are not as affected by a high intake of alcohol

Initial Data Driven Insights

Insight 2



Higher BMI seems to suggest higher Life Expectancy which does not make a lot of sense...

?

Mean BMI is found to be at 38.38, min BMI @ 1.00 and max BMI @ 87.30

!

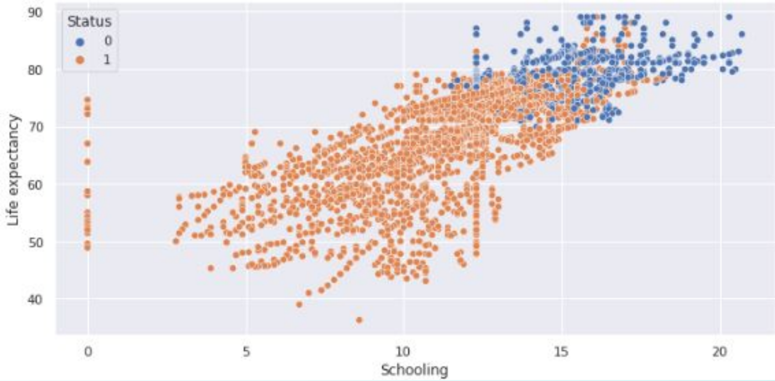
Possible error in the scrapping of data from the WHO site

BMI
2938.00
38.38
19.94
1.00
19.40
43.50
56.10
87.30

What we will do:

Drop BMI to prevent it from affecting the accuracy of our models.

Initial Data Driven Insights



Insight 3

Strong positive correlation between schooling and life expectancy

Strong positive correlation between income composition of resources and life expectancy

How our EDA helped us plan our Regression Models used

Variables with a **correlation of > 0.3** with 'Life Expectancy' was chosen as predictors for our regression models.

Correlation of 0.56 but relationship is **clearly not completely linear!**

SKLearn Linear Regression

Random Forest Regression (Non-Linear)

TensorFlow with the 'Relu' Activation Layer

03. Machine Learning

***"Predicting the future isn't magic,
it's artificial intelligence."
-Dave Waters***

Multivariate Linear Regression Model (SKLearn)

Train-Test-Split

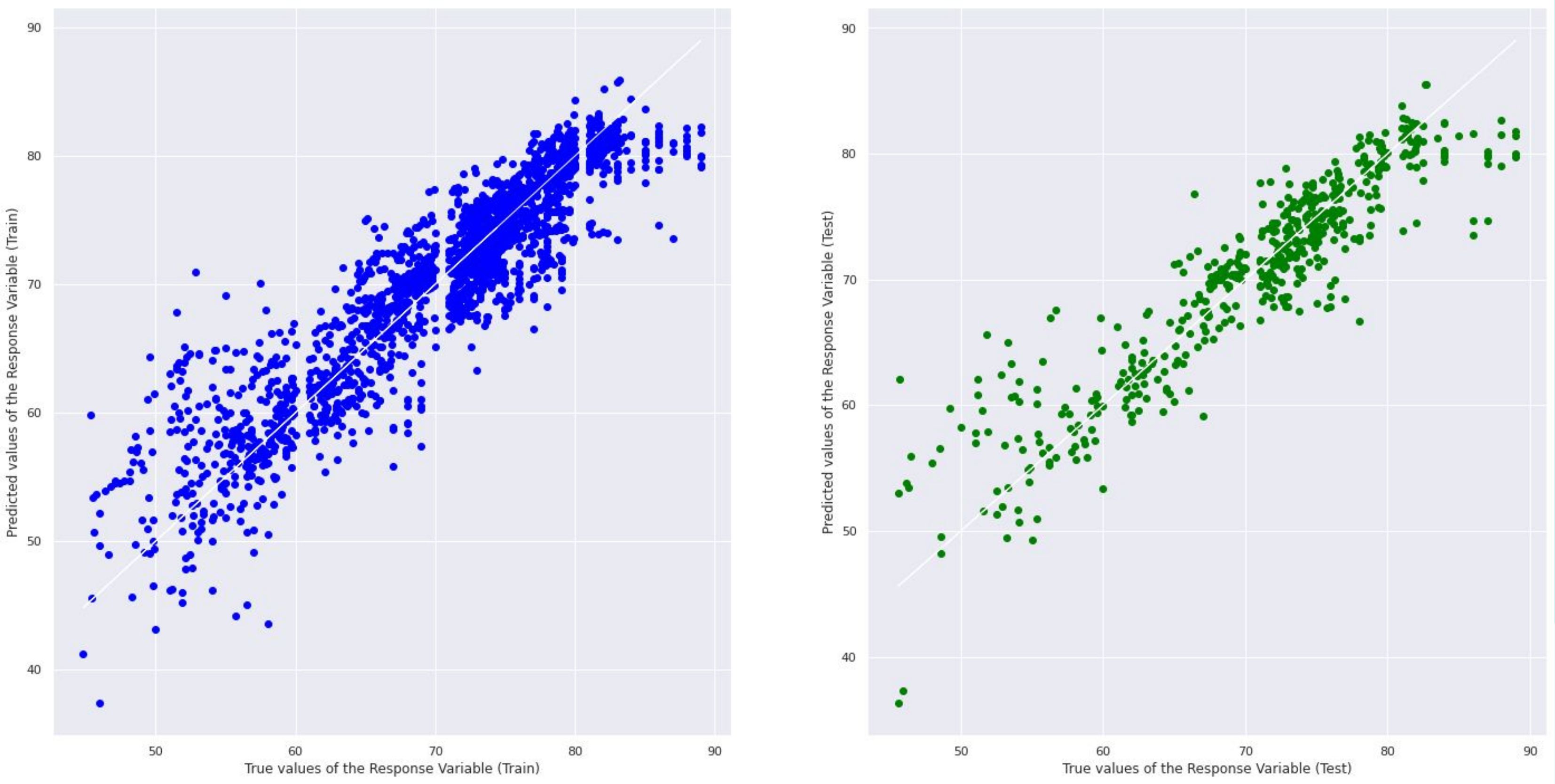
Train Data (80%)	Test Data (20%)

```
linreg = LinearRegression()
linreg.fit(x_train, y_train)
```

Fitting Linear Regression

	Predictors	Coefficients
0	Status	-0.885219
1	Adult Mortality	-0.017002
2	Alcohol	-0.023813
3	percentage expenditure	0.000328
4	Polio	0.016877
5	Diphtheria	0.034610
6	HIV/AIDS	-0.630223
7	Schooling	-0.154049
8	Income composition of resources	33.320929
9	GDP	-0.000035
10	thinness 10-19 years	0.064694
11	thinness 5-9 years	-0.282373

Multivariate Linear Regression (Train & Test)



Goodness Of Fit Of Model

Train Dataset

Explained Variance (R^2): 0.842155

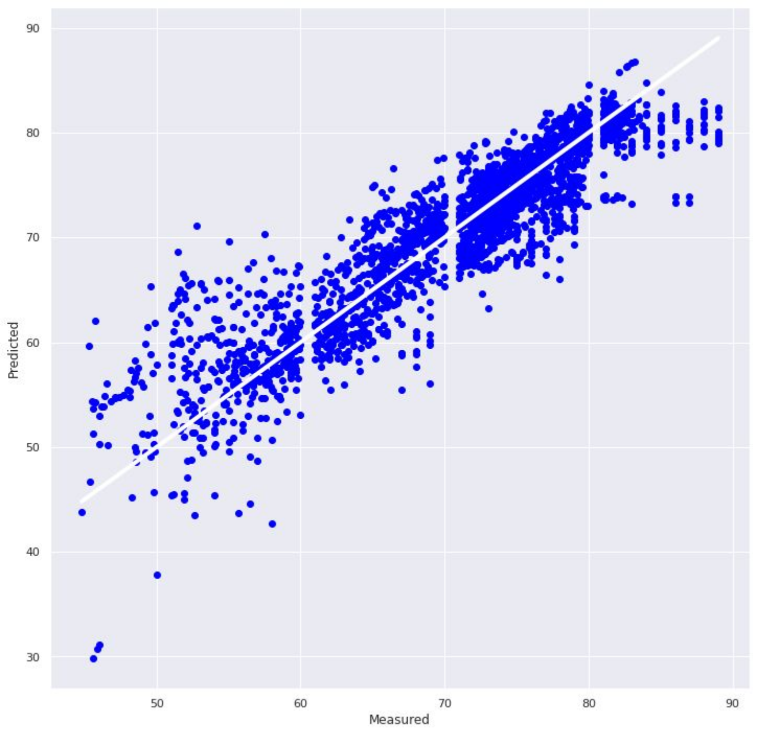
Mean Squared Error (MSE): 12.19263

Test Dataset

Explained Variance (R^2): 0.820759

Mean Squared Error (MSE): **13.489578**

10-Fold Cross Validation




```
scores = cross_val_score(model, x_factors,
y_factor,scoring= 'neg_mean_squared_error' ,cv=cv
, n_jobs=-1)
```

Using MSE as our scoring

```
Mean Squared Error:
12.632429219107204
```

MSE using 10-fold Cross Validation

Random Forest Regression Model (Ensemble)

Train-Test-Split

Train Data (80%)	Test Data (20%)

Determining best number of estimators

Code:

```
regressor = RandomForestRegressor(n_estimators=100, random_state=0)
CV_rfc = GridSearchCV(estimator=regressor, param_grid=param_grid, cv= 5)
```

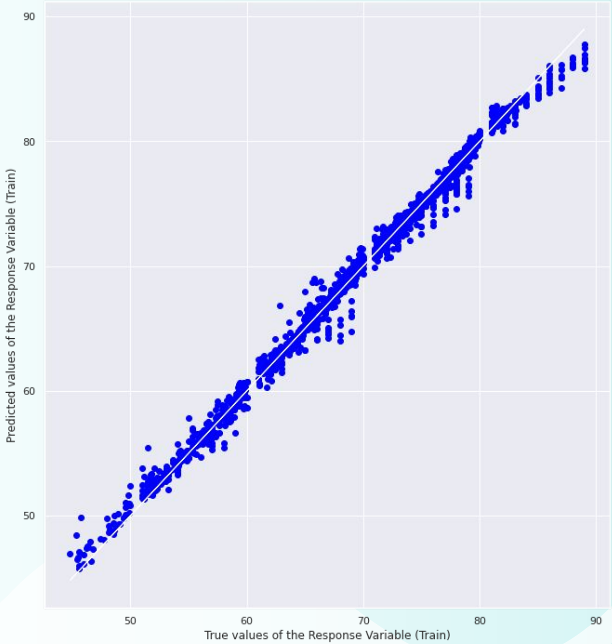
Result:

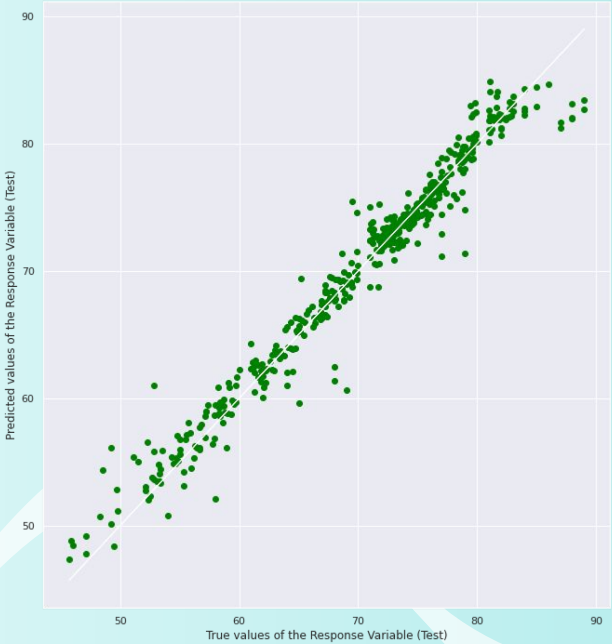
By GridSearch, we have determined that the best number of estimators is **101**

```
regressor =
RandomForestRegressor(n_esti
mators=101, random_state=1)
regressor.fit(X_train,
y_train.values.ravel())
```

Fitting the model using 101 estimators

Random Forest Regression Model (Ensemble)

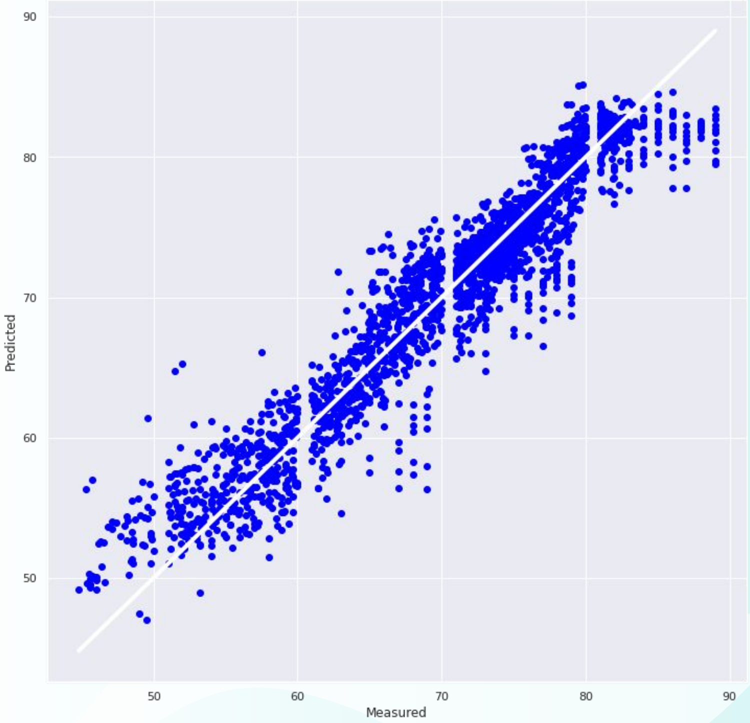




MSE: 2.84579833835703

Overfitting?

10-Fold Cross Validation




```
scores = cross_val_score(model, x_factors,
y_factor,scoring='neg_mean_squared_error',cv=cv
, n_jobs=-1)
```

Using MSE as our scoring

```
Mean Squared Error:
3.2996310929365658
```

MSE using 10-fold Cross Validation

Deep Neural Network (TensorFlow)

Replacing spaces with dashes to prepare for regression with TensorFlow

```
Data columns (total 12 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Status                              2578 non-null   int64
 1   Adult_Mortality                     2578 non-null   float64
 2   Alcohol                             2578 non-null   float64
 3   percentage_expenditure              2578 non-null   float64
 4   Polio                               2578 non-null   float64
 5   Diphtheria                          2578 non-null   float64
 6   HIV/AIDS                            2578 non-null   float64
 7   Schooling                           2578 non-null   float64
 8   Income_composition_of_resources     2578 non-null   float64
 9   GDP                                 2578 non-null   float64
 10  thinness_10-19_years                2578 non-null   float64
 11  thinness_5-9_years                  2578 non-null   float64
```

Deep Neural Network (TensorFlow)

Train-Test-Split

Train Data (80%)	Test Data (20%)

```
standard_scaler = StandardScaler()
```

Standard scale test and train variables

Tensorflow Sequential Model

```
def build_model_using_sequential():
  model = Sequential([
    Dense(hidden_units1, kernel_initializer='normal',
activation='relu'),
    Dropout(0.2),
    Dense(hidden_units2, kernel_initializer='normal',
activation='relu'),
    Dropout(0.2),
    Dense(hidden_units3, kernel_initializer='normal',
activation='relu'),
    Dense(1, kernel_initializer='normal',
activation='linear')
  ])
  return model
```

Deep Neural Network (TensorFlow)

```
mse = MeanSquaredError()
model.compile(
loss=mse,
optimizer=Adam(learning_rate=learning
_rate),
metrics=[mse]
)
```

Loss Function

```
earlystopping =
callbacks.EarlyStopping(monitor
="val_loss",

mode ="min",
patience = 2,
restore_best_weights = True)
```

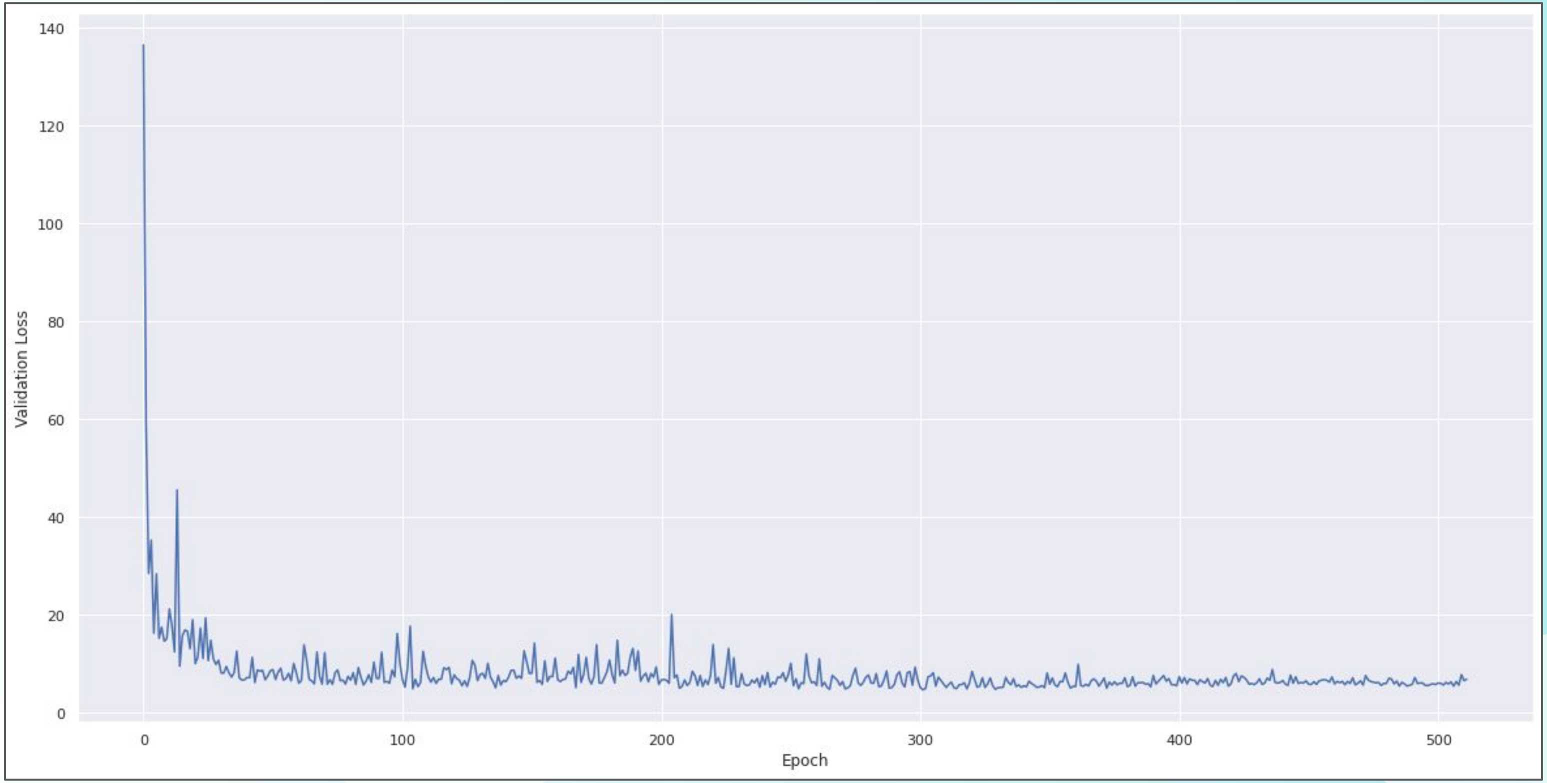
Early Stopping

History

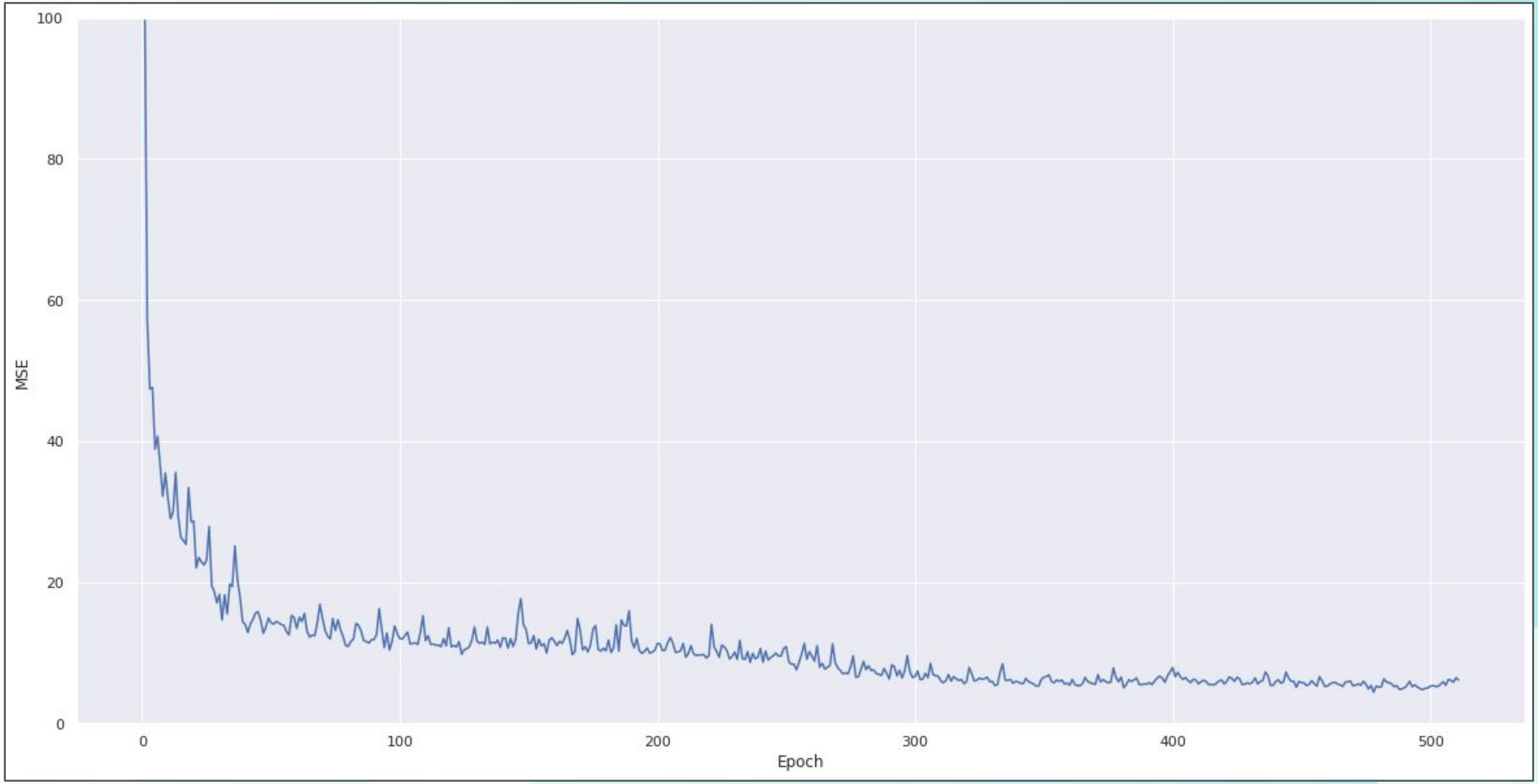
Epochs: 512

Epoch	Val Loss
15	9.4643
256	5.8942
508	5.5905

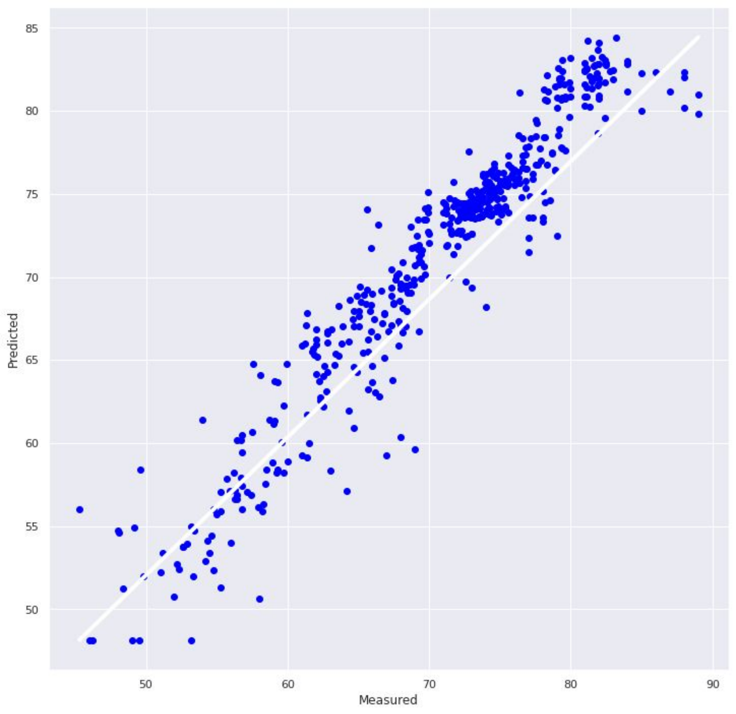
Deep Neural Network (Val Loss vs Epoch)



Deep Neural Network (MSE vs Epoch)



Deep Neural Network



Difference between Predicted & Measured is not that much

MSE: 6.451314449310303

04. Conclusion

“Data is the new science. Big Data holds the answers.” – By Pat Gelsinger

Outcome Of Project

Model	Minimum MSE (2 d.p)
SKLearn Multi-Variate Regression with Cross Validation	12.63
Random Forest Regression with Cross Validation	$\boxed{3.30}$
Multi-variate Regression with TensorFlow	6.45

! **Surprisingly, Random Forest Regression generated the best result using MSE as the metric**

Outcome Of Project

Why was Deep Learning worse off than Random Forest?

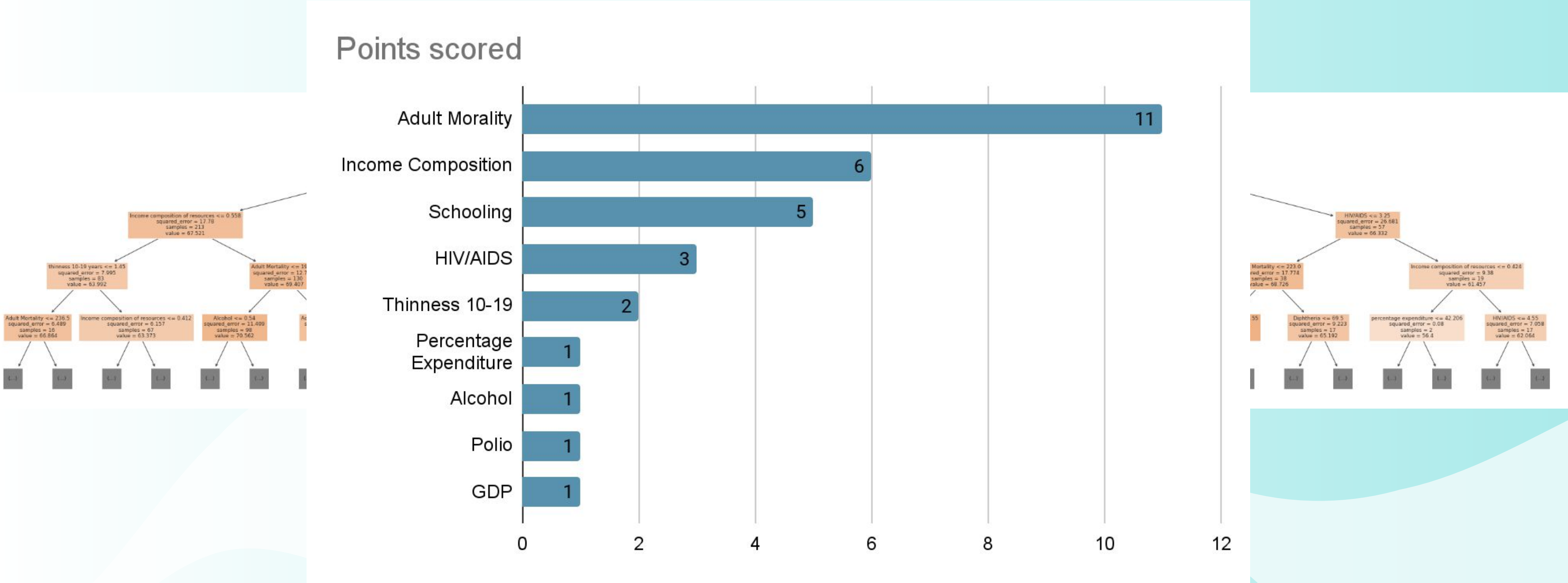
Deep Learning requires extremely large datasets

Size of Dataset

2578 Rows of Data

Deep Learning still performed better than linear regression, suggesting that the relationship between the predictors and Life Expectancy **may not have been linear to begin with.**

Outcome Of Project



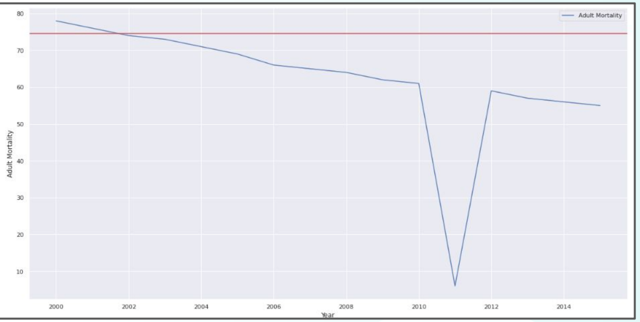
Outcome Of Project

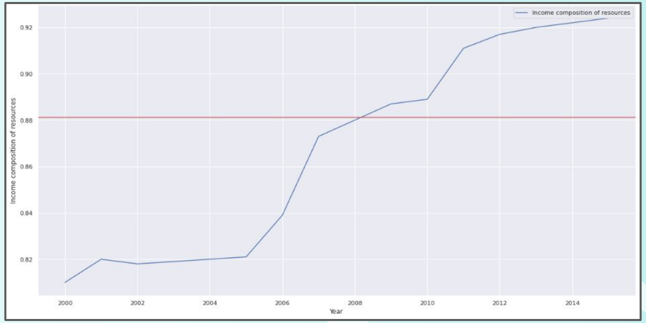
Main areas of concern to prioritise

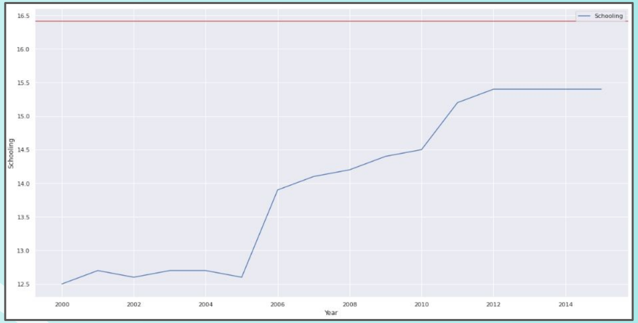
- Adult Mortality
- Income Composition
- Schooling

How to increase Singapore's life expectancy **effectively?**

Outcome Of Project







Adult Mortality

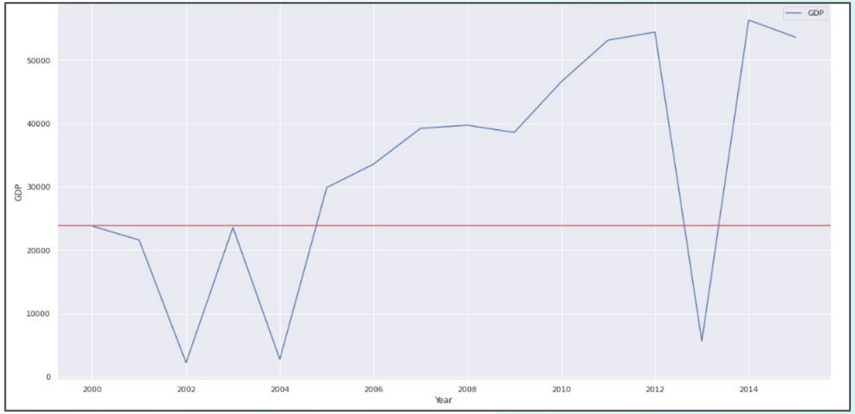
Income

Schooling

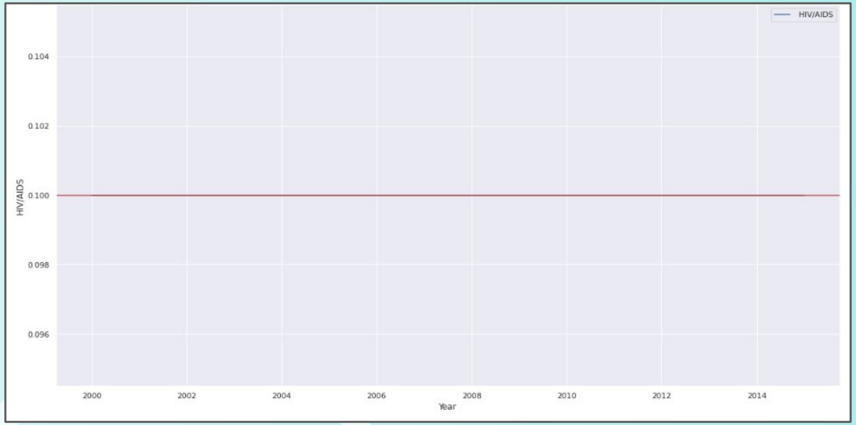
Outcome Of Project

Schooling

Data Driven Insight



GDP



HIV/AIDS

Data Driven Insight

Variable	Value	Life Expectancy	Improvement
Schooling	12.9	83.63	-
Schooling	**13.9**	**84.19**	**+ 0.56**
Income Composition of Resources	0.867	81.55	-
Income Composition of Resources	**0.917**	**81.87**	**+ 0.32**
Adult Mortality	62.0	81.46	-
Adult Mortality	**57.0**	**81.65**	**+ 0.19**

Recommendations

1. Invest more funds and resources to subsidise citizens' higher education to increase years of average schooling

2. Better utilisation of resources (e.g. manpower) allows efficient allocation of resources to healthcare

3. Further improve healthcare services and provide incentives for individuals to lead healthier lifestyles (e.g. Cash rebates for exercising)

THANKS!