DATA1002-CC-L14-G1 Stage 3 Report

SID and Unikeys

SID: 520 423 035 Unikey:slin7370

SID: 520 102 934 Unikey:pwoo5799

SID: 520 246 478 Unikey:obow2075

SID: 520 463 178 Unikey:sspi3010

Introduction

The attribute for inspection in this report is the Gross Domestic Product per capita in USD, using the dataset developed in the previous 2 stages. GDP is a nominal attribute, as such, we will be using difference methods of regression in order to produce predictive models. Therefore to measure the success of these predictive models, we will be using the r-squared value, and root mean squared error, for both in and out of sample data. This will help to determine overfitting, as well as give an idea of how well these models work on unseen data. For most analysis undertaken, data was split into a test and train set using:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1,
random_state = 42)
```

Section A: Model Production and Evaluation

Elastic Net Linear Model (SID: 520 246 478)

For one of our models, we wanted to use a model that had built-in feature selection. Elastic Net Regularisation is a combination of both the LASSO and Ridge regression in order to avoid the weaknesses of both LASSO and Ridge, while also incorporating where they accel. It does this by linearly combining the penalty factors involved in LASSO and Ridge regression. LASSO has a tendency to group highly correlated variables, then using only 1 of these and discarding the others. This is addressed by using Ridge regression's Tikhonov regularisation.

This was generated using model:

```
1   from numpy import average
2   import pandas as pd
3   import sklearn as sk
4   import numpy as np
5   from sklearn.linear_model import ElasticNet
6   from sklearn.datasets import make_regression
7   from sklearn.metrics import r2_score
8   from sklearn.model_selection import KFold
9   from sklearn.metrics import accuracy_score
10  from sklearn.metrics import mean_squared_error
11  from math import sqrt
12  from math import exp
13
14  df = pd.read_csv("integrated.csv")
15  nfolds = 9
16  GDPrange = df["GDP/capita (USD)"].max() - df["GDP/capita (USD)"].min()
```

```
print("Predicting Base GDP:")
regr = ElasticNet(random_state=0, max_iter=1000)

y = df["GDP/capita (USD)"]
X = df.iloc[:,[3,5,18,19,20,21]]

regr.fit(X,y)

print("Feature names:", regr.feature_names_in_)
print("Intercept:", regr.intercept_)
print("Coefficients:", regr.coef_)


predicted = regr.predict(X)
print("In-sample R2 value:", r2_score(y, predicted))
print("In-sample RMSE:", sqrt(mean_squared_error(y, predicted)))
print("In-sample Normalized Deviation:",
    sqrt(mean_squared_error(y, predicted))/GDPrange)


y = df["GDP/capita (USD)"].to_numpy()
X = df.iloc[:,[3,5,18,19,20,21]].to_numpy()

cv = KFold(nfolds)
acc = []
rmses= []
for train_index, test_index in cv.split(X):
    x_train = X[train_index]
    y_train = y[train_index]
    x_test = X[test_index]
    y_test = y[test_index]
    regr.fit(x_train, y_train)
    predicted = regr.predict(x_test)
    acc.append(r2_score(y_test, predicted))
    rmses.append(sqrt(mean_squared_error(y_test, predicted)))


print("Average R2 value out-of-sample:", average(acc))
print("Average RMSE out-of-sample:", average(rmses))
print("Out-of-sample Normalized Deviation:", average(rmses)/GDPrange)


```

```
print("\nPredicting Logged GDP:")

regr = ElasticNet(random_state=0, max_iter=1000)
y = np.log(df["GDP/capita (USD)"])
X = df.iloc[:,[3,5,18,19,20,21]]

regr.fit(X,y)

print("Feature names:", regr.feature_names_in_)
print("Intercept:", regr.intercept_)
print("Coefficients:", regr.coef_)


predicted = regr.predict(X)
print("In-sample R2 value:", r2_score(y, predicted))
print("In-sample RMSE (in units USD):",
    exp(sqrt(mean_squared_error(y, predicted))))
print("In-sample Normalized Deviation:",
    exp(sqrt(mean_squared_error(y, predicted)))/GDPrange)

y = np.log(df["GDP/capita (USD)"]).to_numpy()
X = df.iloc[:,[3,5,18,19,20,21]].to_numpy()

cv = KFold(nfolds)
acc = []
rmses = []
for train_index, test_index in cv.split(X):
    x_train = X[train_index]
    y_train = y[train_index]
    x_test = X[test_index]
    y_test = y[test_index]
    regr.fit(x_train, y_train)
    predicted = regr.predict(x_test)
    acc.append(r2_score(y_test, predicted))
    rmses.append(sqrt(mean_squared_error(y_test, predicted)))

print("Average R2 value out-of-sample:", average(acc))
print("Average RMSE out-of-sample (in units USD):", exp(average(rmses)))
print("Out-of-sample Normalized Deviation:",
    exp(average(rmses))/GDPrange)
```

Results for the Elastic Net method are as follows:

```
Predicting Base GDP:
Feature names: ['Life Expectancy (Years)'
 'Deaths - Interpersonal violence - Sex: Both - Age: All Ages (Rate/100000)'
 'Average Years of Schooling Attained'
 'Average Years of Primary Schooling Attained'
 'Average Years of Secondary Schooling Attained'
 'Average Years of Tertirary Schooling Attained']
Intercept: -66314.3390477161
Coefficients: [1189.23934973 -316.92987509 -491.53701105 1130.36241406 -836.29862218
 -770.55208997]
In-sample R2 value: 0.42083259588488064
In-sample RMSE: 14531.781152305606
In-sample Normalized Deviation: 0.13123106268443305
Average R2 value out-of-sample: 0.33637956771526867
Average RMSE out-of-sample: 14562.885520134296
Out-of-sample Normalized Deviation: 0.1315119545586985

Predicting Logged GDP:
Feature names: ['Life Expectancy (Years)'
 'Deaths - Interpersonal violence - Sex: Both - Age: All Ages (Rate/100000)'
 'Average Years of Schooling Attained'
 'Average Years of Primary Schooling Attained'
 'Average Years of Secondary Schooling Attained'
 'Average Years of Tertirary Schooling Attained']
Intercept: -0.6981187062841023
Coefficients: [ 0.13070432 -0.          0.          0.          0.          -0.        ]
In-sample R2 value: 0.6660569548806029
In-sample RMSE (in units USD): 2.543024309944142
In-sample Normalized Deviation: 2.2965098299279596e-05
Average R2 value out-of-sample: 0.5709697764899686
Average RMSE out-of-sample (in units USD): 2.5829726720251505
Out-of-sample Normalized Deviation: 2.3325856967019477e-05
```

Firstly two models were created. Based on information regarding the relationship between a variety of the features and GDP, a logged model, as well as a base model. For both models, the input features were: life expectancy, deaths from interpersonal violence, average years of schooling, average years of primary schooling, secondary schooling, and tertiary schooling. For the base model, elastic net used all features and tested as such:

Base Model	**R2**	**RMSE**	**Norm RMSE**
In-Sample	0.42	14531.78	0.13
Out-Sample	0.34	14562.89	0.13

This model performed ok, however with an r-squared of 0.34 out of sample, its predictions aren't particularly reliable. Root mean squared error remains relatively the same for both in sample and out of sample testing, being about a 13% difference ($14500~ USD). Additionally, we see little evidence of overfitting or underfitting as the r-squared values are

relatively similar, as well as RMSE and Norm RMSE. Alternatively, using a Logged GDP model we have:

Logged Model	R2	RMSE (USD)	Norm RMSE
In-Sample	0.67	2.54	2.29E-05
Out-Sample	0.57	2.58	2.33E-06

For the Logged Model, elastic net selected only life expectancy as a predictor, resulting in a higher out of sample r-squared of 0.57. RMSE was converted to USD, and was around $2.54 USD, which is very low compared to the range. Again, as the r-squared are relatively similar, there is little evidence of overfitting or underfitting as RMSE and Norm RMSE are also very similar.

Linear Regression by Death Rate (SID: 520 102 934)

For this section, we will be using linear regression modelling. We will be using the death rate column of our dataset as a predictor for our GDP column. We decided to use a linear regression model because the graph somewhat resembles a negative logarithmic graph yet not well suited enough to use a logarithmic regression model.

The code is used to get the mode is the following:

```python
import pandas as pd
from math import sqrt
from sklearn import linear_model
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
import numpy as np

df = pd.read_csv('integrated.csv')
X = np.log(df[['Deaths - Interpersonal violence - Sex: Both - Age: All Ages (Rate/100000)']])
y = df["GDP/capita (USD)"]
RangeofGDP = df["GDP/capita (USD)"].max() - df["GDP/capita (USD)"].min()

# linear regression for actual data
regr = linear_model.LinearRegression().fit(X,y)

print("Feature Name: " + str(regr.feature_names_in_))
print("Intercept: " + str(regr.intercept_))
print("Coefficients: " + str(regr.coef_))

prediction = regr.predict(X)
inr2score = metrics.r2_score(y, prediction)
print("In-sample R2 value: ", inr2score)
inrmse = sqrt(metrics.mean_squared_error(y , prediction))
print("In sample RMSE: ", inrmse)
innorm = sqrt(metrics.mean_squared_error(y, prediction))/RangeofGDP
print("In-sample Normalized Deviation: ", innorm)

print("\n")
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
# linear regression for train data
regr = linear_model.LinearRegression().fit(X_train,y_train)

# Use the model to predict y from X_test
y_pred = regr.predict(X_test)
# R-squared score: 1 is perfect prediction
r2score =  metrics.r2_score(y_test, y_pred)
print('Out-of-sample R2 value', r2score)
# Root mean squared error
rmse = sqrt(metrics.mean_squared_error(y_test, y_pred))
print('Out-of-sample RMSE:', rmse)
# Normalized Deviation
NDev = rmse/RangeofGDP
print('Out-of-sample Normalized Deviation: ', NDev)

```

This is the output of the code that shows the results using linear regression method:

```
Feature Name: ['Deaths - Interpersonal violence - Sex: Both - Age: All Ages (Rate/100000)']
Intercept: 26466.725410617546
Coefficients: [-9033.35224684]
In-sample R2 value:  0.308957708994301
In sample RMSE:  15873.368784376957
In-sample Normalized Deviation:  0.1433464371726514


Out-of-sample R2 value 0.35023842063484234
Out-of-sample RMSE: 13024.038700619467
Out-of-sample Normalized Deviation:  0.1176152063681678
```

At the start, we were going to use np.log(GDP column). However, according to the r-test results, it wasn't well suited compared to taking the log of the death rate column. We also tried to do a negative log of the GDP column but it still didn't improve the model.

Logged Model	R2	RMSE	Norm RMSE
In-Sample	0.31	15873.37	0.14
Out-of-Sample	0.35	13024.04	0.12

This model performed fine. It has a r-squared value of 0.35 out of sample meaning the predictions aren't particularly reliable. The root mean squared error decreased by around 2800 USD in out of sample compared to in sample. The out of sample has a 12 percent difference, about 13000 USD. Since the r-squared values are similar, there is little evidence of overfitting or underfitting, even though root mean squared error is a bit lower in out of sample model than in sample model, hinting that out of sample is a better predictor.

Linear Regression by Years of Schooling (SID: 520423035)

To predict the GDP/capita from Average Years of Schooling Attained, a Linear Regression model was built. The linear regression model attempts to model the relationship between 2 quantitative variables by fitting a linear line that attempts to minimise the residual sum of squares between the 2 variables. In this model, the independent variable is “Average Years of Schooling Attained” and the dependent variable is “GDP/capita (USD)”.

Building the model:

```
import pandas as pd
from sklearn import linear_model
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn import neighbors
import math
import numpy as np

df = pd.read_csv("integrated.csv")      #Reading in data

data = df[df["Year"] == 2010]           #Taking only data values in 2010

data['Log (Base 10) of GDP/capita (USD)'] = np.log10(data['GDP/capita (USD)'])      #Adding a column for log base 10 of GDP/capita (USD)

X = data[["Average Years of Schooling Attained"]]   #Determining independent variable and grabbing its data
Y = data[["GDP/capita (USD)"]]                      #Determining dependent variable and grabbing its data

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.1, random_state = 42)   #Splitting data into training and testing sets

linear_nonlog = linear_model.LinearRegression().fit(X_train, Y_train)   #Creating model with training data

out_of_sample = X_test
in_sample = X_train
```

To build the model, we first split our data (which is read and stored in a pandas dataframe) into training and testing sets. X stands for the independent variable and train stands for the training set whereas test stands for the testing set. The test_size was set to 0.1 which meant 10% of the data was selected to do testing (90% was used to test). We then created the model called linear_nonlog by using the sklearn Linear Regression model and fitting the best fit line to the training data.

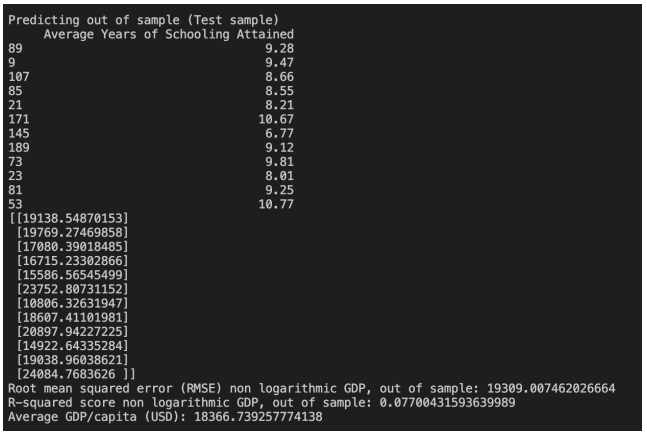
Code to find results:

```
#NON LOGARITHMIC

#Out of sample test
print("Predicting out of sample (Test sample)")
linearPredict = linear_nonlog.predict(out_of_sample)
print(out_of_sample)    #Printing test sample and also the results of prediction
print(linearPredict)

mse = metrics.mean_squared_error(Y_test, linearPredict)
print('Root mean squared error (RMSE) non logarithmic GDP, out of sample:', math.sqrt(mse))
print('R-squared score non logarithmic GDP, out of sample:', metrics.r2_score(Y_test, linearPredict))    #Out of sample
print()

#In sample test
print("Predicting in sample (Training sample)")
linear_predict_in_sample = linear_nonlog.predict(in_sample)
print(in_sample)
print(linear_predict_in_sample)

mse_in_sample = metrics.mean_squared_error(Y_train, linear_predict_in_sample)
print("Root mean squared error(RMSE) non logarithmic GDP, in sample: ", math.sqrt(mse_in_sample))
print("R-squared score non logarithmic GDP, in sample: ", metrics.r2_score(Y_train, linear_predict_in_sample))

print()
print("Coefficient")
print(linear_nonlog.coef_) #Coeff of 3319 means for each increase in one year of average years of schooling, GDP/capita (USD) is expected to increase by 3319$
```

The predictions were done by using our linear_nonlog model to first predict our testing set (out of sample). The data fed into the model and the predictions were then printed. The

same was done for the training set (in sample). The coefficient of the model was then also printed.

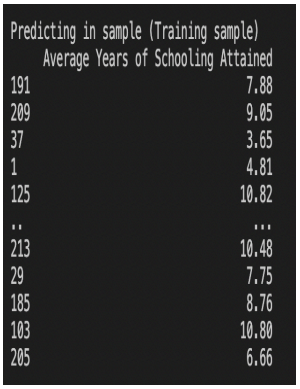
Results:

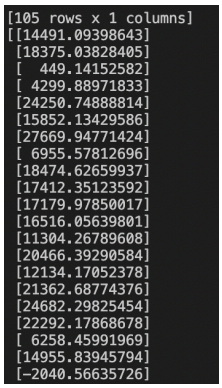


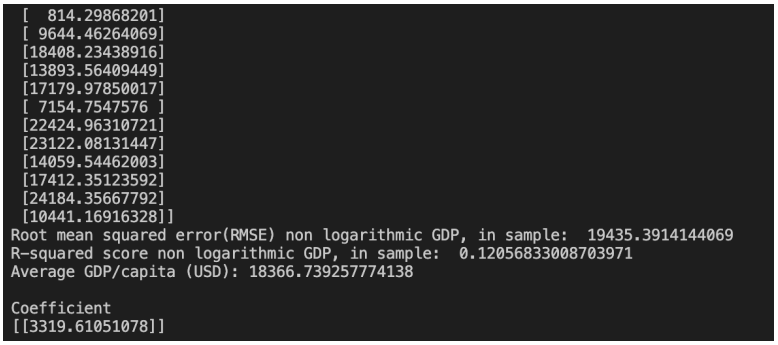
Out of sample test:

The testing set predicted with a R-squared score of 0.077 which is very low and suggests the predictions were not accurate at all. The RMSE was also very high compared to the mean of GDP/capita (USD) of the test sample which was $18366. The RMSE was larger than the mean GDP/capita which suggests there is a lot of uncertainty with the predictions.

In sample test:



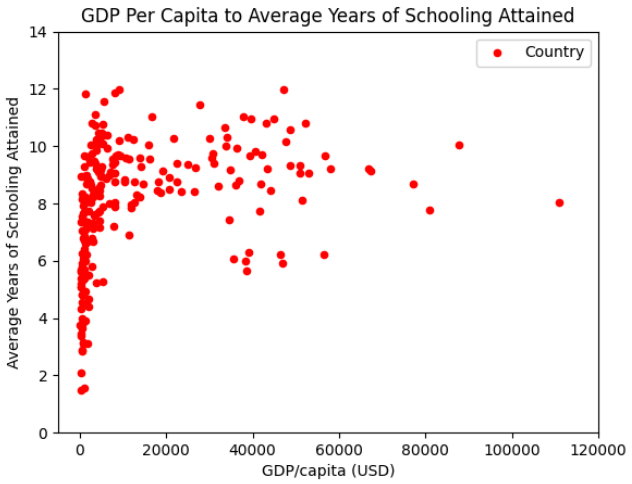




The predictions were run again using the training sample. The R-Squared score of the training sample was 0.12 which again is rather low. It is higher than the R-Squared score by 50% which may suggest some overfitting. It is also possible that the R-Squared score increases due to the larger sample size of the inputs. However the R-Squared score for both

are too low to deem this model effective. Again the RMSW is rather large compared to the average GDP/capita of the training sample.
The coefficient also shows the relationship between Average Years of Schooling Attained and GDP/capita (USD). In this model, it suggests that for each increase in year of Average Years of Schooling Attained, GDP/capita (USD) is expected to increase by $3319.

If we were to visualise the graph of GDP/capita and Average Years of Schooling Attained,



We can clearly see that the relationship between the 2 attributes are not linear. This may explain the low R-Squared scores for the 2 predictions and also suggests that this model is not valid and usable for a linear regression prediction.

To account for the logarithmic relationship between GDP/capita and Average Years of Schooling Attained, the model was built again but this time having the dependent variable to be the log base 10 of GDP/capita (USD).

```
82
83  Y_log = data[['Log (Base 10) of GDP/capita (USD)']]
84  X_train, X_test, Y_logtrain, Y_logtest = train_test_split(X, Y_log, test_size = 0.1, random_state = 42)
85  linear_log = linear_model.LinearRegression().fit(X_train, Y_logtrain)
86  out_of_sample = X_test
87  in_sample = X_train
88
89
90  print("Predicting out of sample (Test sample)")
91  linearPredictLog = linear_log.predict(out_of_sample)
92  mse_log = metrics.mean_squared_error(Y_logtest, linearPredictLog)
93  print('Root mean squared error (RMSE) logarithmic GDP, out of sample:', math.sqrt(mse_log))
94  print('R-squared score logarithmic, in sample:', metrics.r2_score(Y_logtest, linearPredictLog))  #Out of sample
95  print("Average Log of GDP/capita (USD): " + str(linearPredictLog.mean()))
96
97  print()
98  print()
99
100 print("Predicting in sample (Training sample)")
101 linearPredictLog_inSample = linear_log.predict(in_sample)
102 mse_logInSample = metrics.mean_squared_error(Y_logtrain, linearPredictLog_inSample)
103 print("Root mean squared error (RMSE) logarithmic, in sample:" , math.sqrt(mse_logInSample))
104 print("R-Squared score, logarithmic GDP, out of sample: ", metrics.r2_score(Y_logtrain, linearPredictLog_inSample))
105 print("Average Log of GDP/capita (USD): " + str(linearPredictLog_inSample.mean()))
106
107 print()
108 print("Coefficient")
109 print(linear_log.coef_)
```

```
Predicting out of sample (Test sample)
Root mean squared error (RMSE) logarithmic GDP, out of sample: 0.548876965206786
R-squared score logarithmic, in sample: 0.24556397593856816
Average Log of GDP/capita (USD): 3.937016305928262


Predicting in sample (Training sample)
Root mean squared error (RMSE) logarithmic, in sample: 0.5200218129339405
R-Squared score, logarithmic GDP, out of sample:  0.3856638212314626
Average Log of GDP/capita (USD): 3.743661779838128

Coefficient
[[0.1900646]]
```

For the out of sample test:
The R-Squared value of the test sample is now at 0.24556 which is still low for the model to be considered effective. However the RMSE value is proportionally a lot lower than before.

For the in sample test:
The R-Squared value of the training sample is now at 0.38566 which is about 50% higher than the out of sample test and may suggest overfitting. However again it is also possible that this difference is due to the increased sample size.The RMSE value is roughly similar compared to the out of sample test.

Summary Evaluation:
Overall, the R-Squared value for both models (Logarithmic and Non-Logarithmic) are quite low and hence suggests the predictions are not accurate. The RMSE values for the non-logarithmic model is very high (higher than the actual mean) which suggests large uncertainty in the predictions. This may be due to the nature of the relationship being exponential rather than linear.
As the relationship between the 2 quantitative attributes is logarithmic, a linear regression model accounting for this was built. For the logarithmic model, the R-Squared values are higher than the non-logarithmic models but still quite low to deem the model acceptable and accurate. The proportion of the RMSE to the mean is significantly smaller than the RMSE for the non logarithmic model which suggests lower uncertainty in the predictions.

For both the models, the R-Squared score is proportionately significantly higher for the training set compared to the testing set which can suggest overfitting.

As the R-Squared values are still low for both approaches, it means Average Years of Schooling does not explain the observed value well and there may be other factors that are more indicative of GDP.

Linear Regression for correlation by Life Expectancy (SID: 520463178)

This model tested the relationship between a country's life expectancy and GDP per capita using linear regression. Linear regression is an analytical method used to model the relationship between two variables, allowing for estimates of values using a line of best fit. In

this model the independent variable is the life expectancy of a country and the dependent variable is the GDP per capita of that same country.

```python
import pandas as pd
import sklearn as sk
import numpy as np
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from math import sqrt


df = pd.read_csv('integrated.csv')

x = df[["Life Expectancy (Years)"]]
y = df[['GDP/capita (USD)']]

X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size = 0.1, random_state = 42)
regr = linear_model.LinearRegression()
regr.fit(X_train, Y_train)

print("Coefficients:", regr.coef_)

y_pred = regr.predict(X_test)
mse = mean_squared_error(Y_test, y_pred)

print('Root mean squared error (RMSE):', sqrt(mse))
print('R-squared score:', r2_score(Y_test, y_pred))
```

The main structure of the code was taken from the provided grok module on linear regression. The first part that was altered was the dataset ‘integrated.csv’, containing all the group’s cleaned data from stages 1 and 2 which was imported using the ‘read’ function. The x axis of the plot was the life expectancy of each country in years, representing the independent variable and the y axis assigning the dependent variable of GDP per capita. The data was then split into training sets and testing sets, the letter X and Y representing their respective axes. Using the linear model created with the regr.fit function, the coefficients, root mean squared error (RMSE) and R-squared score were calculated. The values are given in the image below

```
Coefficients: [[1199.05649727]]
Root mean squared error (RMSE): 10917.402570400058
R-squared score: 0.543436360629064
```

Evaluation:

The model produced was relatively effective as the R-squared score was relatively high at 0.54 (2 d.p) suggesting that there was a fairly high correlation between the variables. This value suggests that there is some correlation between a country’s life expectancy and its GDP per capita and a strong overall linear relationship between the two variables. The root mean squared error when normalised came to around 0.09 which is a relatively low value and suggests only a small margin of error and a high degree of certainty in the predictions of the model. Overall, considering the values obtained, this model was successful to a high degree as it showed a strong correlation between the two variables and a reasonable level of certainty in its predictions.

Section B: Conclusions

Elastic Net Linear Model (SID: 520 246 478)

The logged model produced by the elastic net technique performed reasonably well, with an r-squared value of around 0.57. This was an improvement to the base model, with an r-squared value of 0.34 and solved some of the issues regarding the nature of the collected data. The reason the logged model performed substantially better is due to the non-linear relationship between GDP and life expectancy. By taking the log of GDP, we can find the relationship between log(GDP) and life expectancy to see that this is a much more linear relationship. This also applied for the other features, however the model's internal feature selection only selected life expectancy for this model, because the fit was best. One weakness of the elastic net method however is that these features are selected based on an algorithm, specifically, AIC, BIC, and r^2 cross validation. While this may produce an accurate model, it may make poor choices in feature selection from a domain knowledge point of view. This was controlled by inputting a limited number of features into the elastic net algorithm, and therefore, we avoid uninterpretable, or nonsensical feature selection. This method however, because of its feature selection, allows us to produce effective models without the need for extensive domain knowledge, and can be a good starting point in the investigation of further modelling.

Linear Regression Model for Death Rate (SID: 520 102 934)

The logged model of death rate from interpersonal violence column as a predictor for GDP performed well enough. The linear regression model presented the r-squared value of 0.35. This meant the model wasn't performing bad yet not great either as the perfect model will have a r-square value of 1. The linear regression model also presented a root mean squared error of ~13000 USD, which was a 12% difference. This could be considered a fine amount of error but it could definitely do better. One of the reasons why this model didn't perform well was because the graph of death rate vs GDP looked like a negative logarithm graph. Taking the log of death rate and modelling a linear line performed better than not taking the log of the variables at all. However, it still wasn't enough to make it fit better on the graph. Moreover, there are some extreme values in the dataset and linear regression models are sensitive to outliers. Some modelling methods that allow individual points to be picked to model regressions would be better for this graph.

Linear Regression Model for Education (SID: 520423035)

Overall, the Linear Regression model for education performed quite poorly. Regardless of the approach to the scaling of data to account for possible non linear relationships, the R-Squared values were quite low (0.077 for out of sample test with non logarithmic GDP, and 0.246 for out of sample test with logarithmic GDP), which suggests that the data does not fit the regression model well. Although taking the log of GDP creates a much more linear relationship, the model is not able to accurately predict GDP based on the inputted education data. The model does well in the sense that it attempts to optimise the regression line by minimising the sum of the squared error terms. It also makes the estimation simple and easy to understand as it fits a best fit linear line with a coefficient that is easily interpreted. However, the linearity aspect of the model can also act as its weakness. For example, if more variables were added to the independent variable, the model calculates the effect of each to be constant, and is unable to account for weighting of different variables at different times. For example, if Average Years of Schooling Attained has a higher coefficient for years 0-5 compared to 5+, the model will only evaluate to a constant and not be able to account for changing weightings of the inputs. Also with our data, the plotted points between Average Years of Schooling Attained become quite clustered in some parts, especially for the logarithmic of GDP. It may be more effective to introduce a KNeighbours model with another independent variable such as Life Expectancy to make accurate predictions.

Linear Regression Model for Life Expectancy (SID: 520463178)

The linear regression model for life expectancy was relatively successful. The R-squared value came out as approximately 0.54 which suggests a relatively strong relationship between the x and y variables. This somewhat subpar correlation could be due to underlying health problems in wealthy countries such as obesity, which causes death at a generally younger age. The independent x axis variable for this model was the country's average life expectancy in years, with the dependent y axis variable as GDP per capita in US dollars. The root mean squared error (RMSE) came to around 10917.4 giving a normalised RMSE of around 0.09 or 9% giving a relatively small margin error. This suggests a relatively high degree of accuracy in the prediction made by the model created. Factors that may have resulted in a relatively average level correlation include outliers, which would have skewed the data and decreased the overall correlation. An improved model could have removed outliers from the dataset before taking a linear regression of the two variables.

Comparisons and Conclusion

From our investigation, the highest performance model on unseen data is the Elastic Net using Logged GDP, with an r-squared value of 0.57. This is very similar to the Linear Logged GDP and life expectancy model, with a r-squared of around 0.54 out of sample. This is likely due to the elastic net choosing only life expectancy as a predictor, effectively making these models equivalent.