DATA1002 Stage 3
Group Report

Group
CC - L21 - G3

Member
510239417, ynie7952
520400070, scao2237
510248073, yzhe5295
520066234, zzha8989

Introduction

This report aims to predict the Infant mortality rate using various models. The models mainly used in this group reports are K-nearest regression predictive model and Linear regression model.

The dataset we used is SDG_goal3_clean, which is provided on Canvas. This dataset includes worldwide significant data such as Maternal mortality ratio, Suicide mortality rate, Neonatal mortality rate etc. In the following analysis, we use these different attributes and models to process the prediction of the Infant mortality rate.

SID: 510239417
Unikey: ynie7952

Stage 3 Model Prediction of Infant Mortality Rate

I. Introduction

The dataset we are using in this report is from https://unstats.un.org/sdgs/dataportal . This dataset has 28 columns and 164 rows. In this report, we are mainly used these attribute as following:

Infant mortality rate (deaths per 1,000 live births)::: BOTHSEX
Health worker density, by type of occupation (per 10,000 population): PHYSICIAN

The attribute that we are going to predict is the "*infant mortality rate (deaths per 1000 live births):::both sex* ". We will use this to predict this attribute by using the KNN predictive model.

There are three reasons why I choose the KNN predictive model. Since the attributes that we are extracted are both quantitative, and these two variables that we are focusing on seem not just nearly linear. And the last reason is that there are some outliers in the attributes. (This question will be covered in the report) Therefore, we will build the k-nearest neighbours' regression for our predictions.

II. Prediction

Firstly, I extract the maternal mortality ratio and the infant mortality rate from this dataset. And I split this data into the training dataset and the testing dataset. Then I create one sample to predict the infant mortality rate. (The raw code is shown below)

```
import pandas as pd
from math import sqrt
from sklearn import metrics
from sklearn import neighbors
from sklearn.model_selection import train_test_split
import seaborn as sns
import matplotlib.pyplot as plt


df = pd.read_csv("SDG_goal3_clean.csv")
X = df[["Health worker density, by type of occupation (per 10,000 population)::PHYSICIAN"]]  # slice dataFrame for input variable
y = df[["Infant mortality rate (deaths per 1,000 live births):::BOTHSEX"]] # slice dataFrame for targeted variable
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=42)
neigh = neighbors.KNeighborsRegressor(n_neighbors=4).fit(X_train, y_train)

sample = [100] # create one sample and predict the infant mortality rate
sample_pred_KNN = neigh.predict([sample])


print('----- Sample case using KNN-----')
print("Health worker density by type of physician: ",sample[0])
print("Predicted infant morality rate: ", int(sample_pred_KNN))
print('-----------------------')
```

We use the KNN predictive model to get the root mean squared error and the R-squared score.

```
y_pred_KNN = neigh.predict(X_test)
mse_KNN = metrics.mean_squared_error(y_test, y_pred_KNN)
# Root mean squared error
print('Root mean squared error (RMSE):', sqrt(mse_KNN))
# R-squared score
print('R-squared score:', metrics.r2_score(y_test, y_pred_KNN))
```

Then I got the output of the prediction as following:

```
----- Sample case using KNN-----
Health worker density by type of physician:  100
Predicted infant morality rate:  3
-----------------------
Root mean squared error (RMSE): 16.327469575996826
R-squared score: 0.21384875954566485
```

The R-squared gives an estimate of the relationship between these dependent variables based on the independent variables. We got the Root mean squared error (RMSE) and the R-squared. The RMSE is 16.327469575996826. That is a quite high value. For the R-squared we got is 0.21384875954566485 which is quite low. It seems there is a very weak relationship between these two variables according to our R-squared score. We draw a scatter plot to view the relationship between these two variables in a more straightforward way.

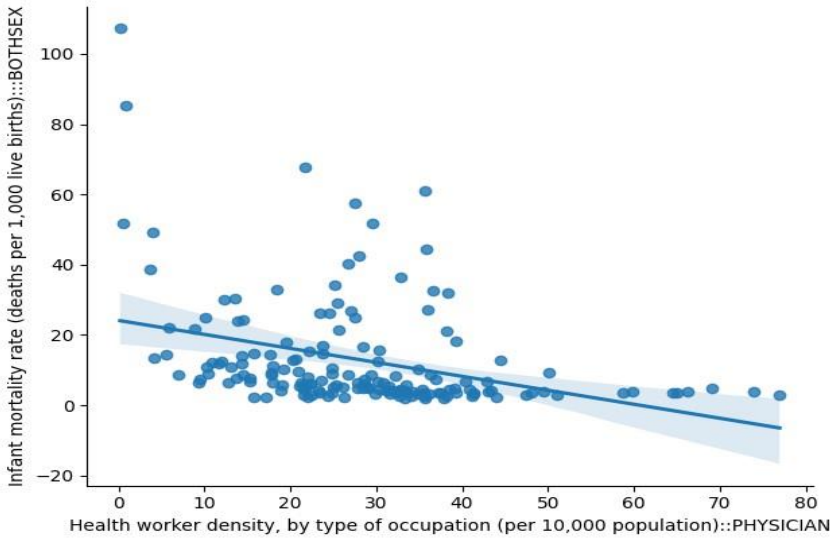


figure1

According to this plot, the value we got by using the KNN predictive model is quite reasonable. It seems that there are some outliers, so we draw another boxplot to have a look at the outliers. The graph as following:

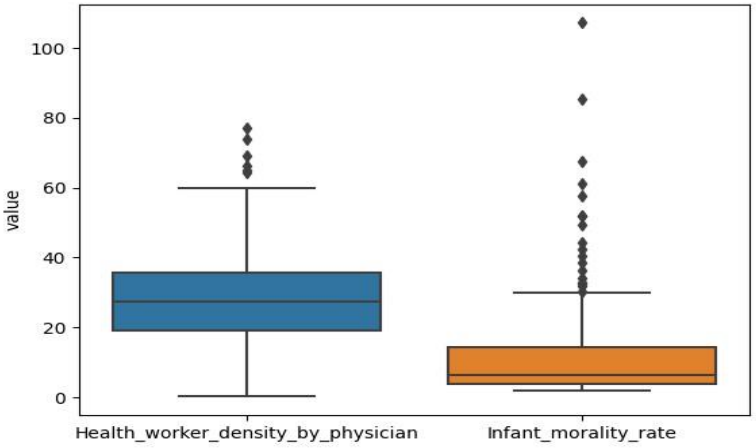


figure2

As you can see, there do have some outliers, especially the infant mortality rate got a lot outliers. Although we have already used the KNN predictive model. There is no way to predict these outliers by using a linear model. In other words, these outliers may have an impact on our result.

III. Model Evaluation

To evaluate our predictive model, we will check the prediction error to verify if our data is overfitting or not. Firstly, we can have a look at the below scatterplot about the test and the prediction.

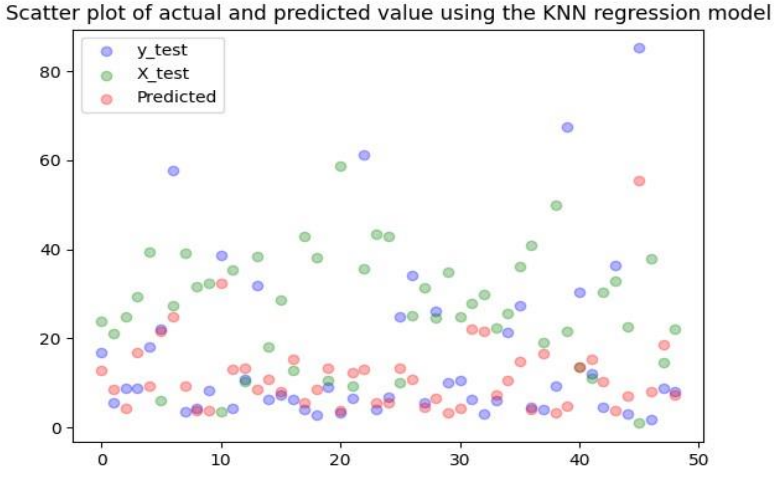


figure3

It seems y test are more likely to fit with the predicted value than x test, however, some outliers here weaken the trend. So, let's take another plot to check the prediction error.

```
## check the prediction error

train_err = {}
test_err = {}
for i in [i for i in range(1, 50)]:
    model = neighbors.KNeighborsRegressor(n_neighbors=i)
    model.fit(X_train, y_train)

    train_prediction = model.predict(X_train)
    train_error = metrics.mean_squared_error(y_train, train_prediction)
    train_err[i]=train_error

    test_prediction = model.predict(X_test)
    test_error = metrics.mean_squared_error(y_test, test_prediction)
    test_err[i]=test_error

plt.plot(list(train_err.keys()), list(train_err.values()),label = "training data error")
plt.plot(list(test_err.keys()), list(test_err.values()),label = "testing data error")
```

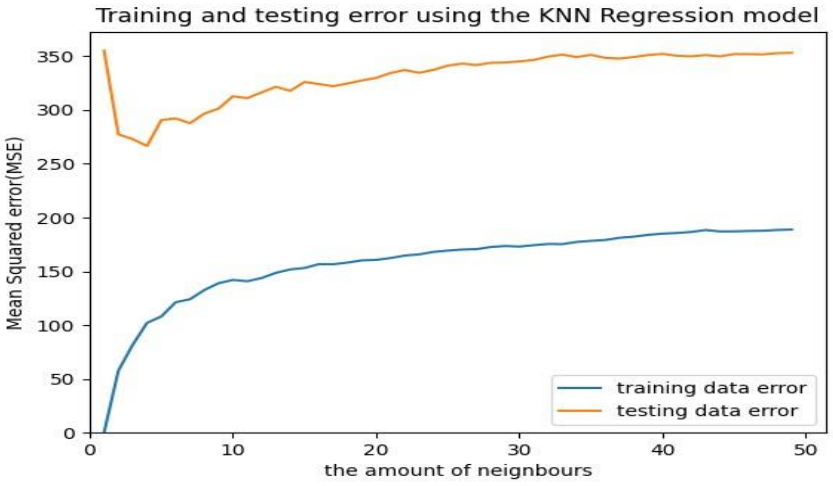


figure4

According to this graph, we can see our model can be considered as an underfitting model, it does not predict well on training dataset. Maybe we should find a richer and more robust model to fit our data.

IV. Conclusion and Limitation

We got a low R-squared score which indicates that there is no strong relationship between these two attributes in terms of the K-nearest neighbours regression. However, our predictive model is not an appropriate choice for this dataset via checking the prediction error. Therefore, our result may not be representative, better predictive models need to be found to predict.

For the limitation, these two variables both got some outliers which may influence the result of the prediction. And the observations of this dataset may not represent the whole population due to the sample limit.

SID: 510248073
Unikey: yzhe5295

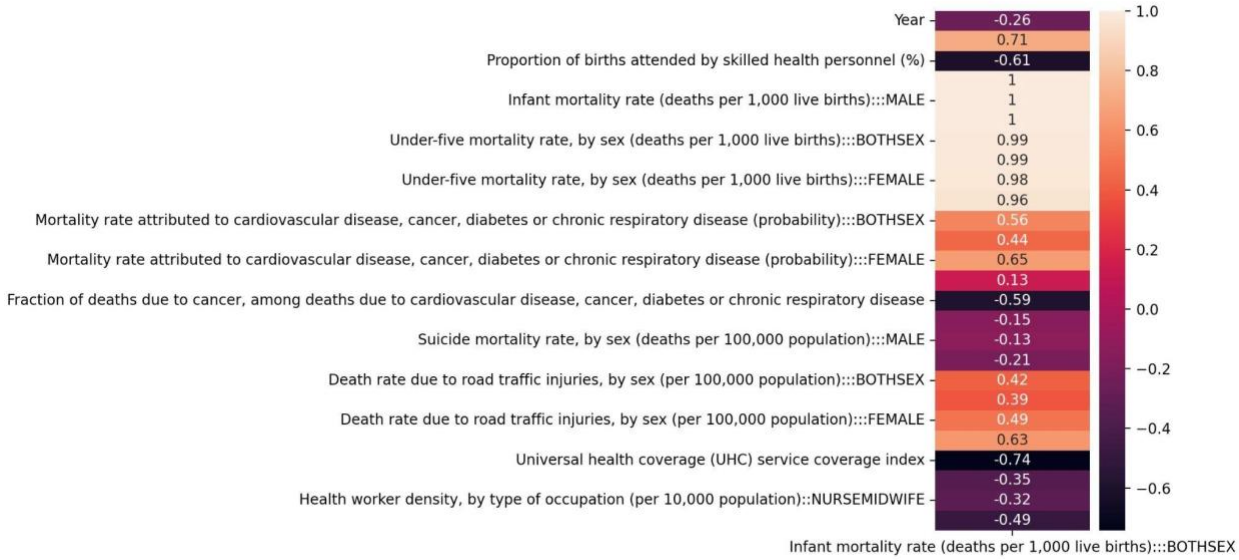
Model Prediction of Infant Mortality Rate

The aim of this report is to produce a model to predict the infant mortality rate. Since the target variable and most of the variables in this dataset are numerical variables, I choose the ***linear regression model*** to produce the following prediction.

Model Selection

As the initial data analysis, I checked the correlation between the target variable Infant mortality and the rest of the variables. The correlation coefficient can express the strength of the linear relationship between two variables. The code and diagram are shown below:

```
dfCorr = df.corr() x = dfCorr[['Infant mortality rate (deaths per 1,000 live births):::BOTHSEX']] sn.heatmap(x, annot=True)
```



(Figure 1: Correlation between Infant mortality rate and other attributes)

According to the visualization, I am able to find out the variables with the highest correlation with the target variable. Thus, I used the following variables as the input variables:

1. *Adolescent birth rate (per 1,000 women aged 15-19 years)*
2. *Universal health coverage (UHC) service coverage index*
3. *Maternal mortality ratio*
4. *Mortality rate attributed to cardiovascular disease, cancer, diabetes or chronic respiratory disease (probability):::BOTHSEX*
5. *Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX*

To access the intercept and the coefficient of the linear regression model, I use the code: `regr.intercept_` and `regr.coef_`. As we can see, the last element, *Under-five mortality rate, by sex,* has the highest absolute coefficient value, which implies that it has more association with the target variable than other attributes.

```
Intercept 0.7972006147588075
```

```
Coefficients:
[-0.00557274 -0.01193058 -0.02224974  0.04871886  0.81724752]
```

Thus, the final model is:

$$Y = 0.7972006 - 0.00557274X_1 - 0.01193058X_2 - 0.2224974X_3 + 0.04871886X_4 + 0.81724752X_5$$

The following code split the dataset into a 10% test set and a 90% train set and applied the linear model. I also create the sample with inputs value to predict the specific infant mortality rate. The code and sample result are shown below:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
regr = linear_model.LinearRegression().fit(X_train, y_train)

sample = [70,50,100,30,50]
sample_pred = regr.predict([sample])
```

```
----- Sample case -----
Adolescent birth rate (per 1,000 women aged 15-19 years): 70
Universal health coverage (UHC) service coverage index: 50
Maternal mortality ratio: 100
Mortality rate attributed to cardiovascular disease, cancer, diabetes or chronic respiratory disease (probability):::
BOTHSEX: 30
Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX: 50
Predicted infant mortality rate: 39
```

Model Evaluation

To evaluate the accuracy of the model, I check the RMSE and the R-squared of the model. As we know that the lower the RMSE, the greater the accuracy of the model, the RMSE of this model is moderate. The R-square of this model is approximately 0.99 and it suggests that approximately 99% of the infant mortality rate can be explained by the model inputs.

```
mse = metrics.mean_squared_error(y_test, y_pred)
print('Root mean squared error (RMSE):', sqrt(mse))
print('R-squared score:', metrics.r2_score(y_test, y_pred))
```

```
Root mean squared error (RMSE): 1.1457444295889523
R-squared score: 0.9936596028876812
```

In sample test is another method that I use to evaluate the accuracy the model. I compare the predicted value with the current dataset, and if the difference between them is less than 2, it is

regarded as a successful prediction. To calculate the accuracy, I use the correct prediction divided by the total numbers of the data. The final accuracy is approximately 92.638%. The code and output are shown below:

```
a = 0.7972006147588111-0.00557274*df['Adolescent birth rate (per 1,000 women aged 15-19 years)'] -0.01193058*df['Universal health coverage (UHC) service coverage index']-0.02224974 *df['Maternal mortality ratio']+ 0.04871886* df['Mortality rate attributed to cardiovascular disease, cancer, diabetes or chronic respiratory disease (probability):::BOTHSEX']+ 0.81724752* df['Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX']
b = df["Infant mortality rate (deaths per 1,000 live births):::BOTHSEX"] i = 0 correct = 0
while i < len(a):
   if abs(a[i]-b[i]) < 2 :
       print(b[i])
       correct +=1
   i += 1

accuracy = correct/len(df) print("In-sample Test
accuracy: ", accuracy*100)
```

```
In-sample Test accuracy:  92.63803680981594
```

Limitation

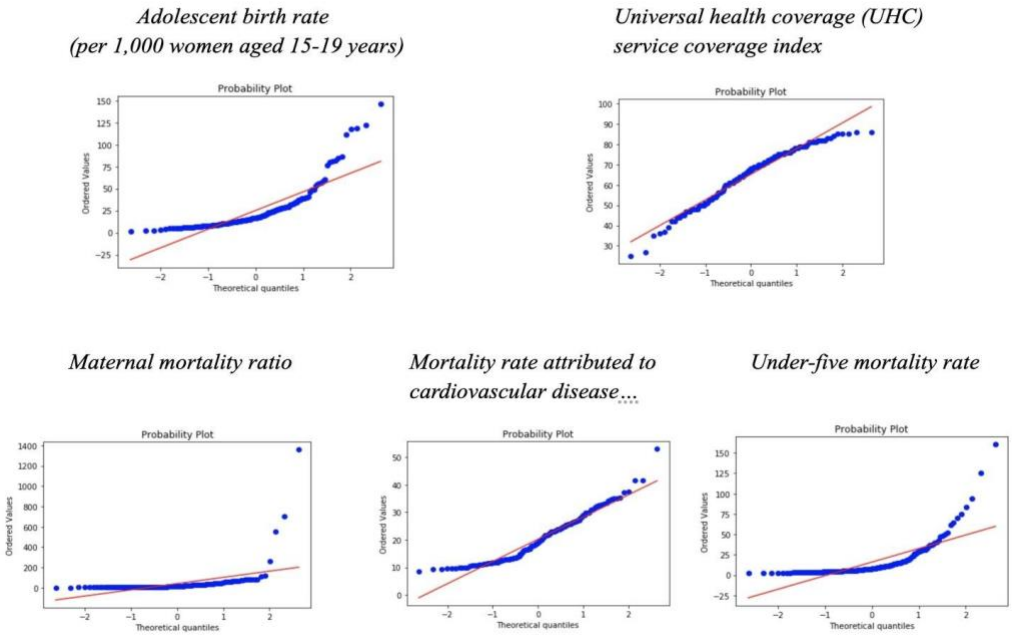
Under-five mortality rate is included Infant mortality rate, which means it is difficult to access the under-five mortality without knowing the infant mortality rate. Thus, it might be unpractical to predict Infant mortality using this attribute.

	Year	Maternal mortality ratio	Proportion of births attended by skilled health personnel (%)	Infant mortality rate (deaths per 1,000 live births):::BOTHSEX	Infant mortality rate (deaths per 1,000 live births):::MALE	Infant mortality rate (deaths per 1,000 live births):::FEMALE	Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX
Year	1.000000	0.016564	0.017481	-0.258015	-0.263810	-0.250120	-0.212285
Maternal mortality ratio	0.016564	1.000000	-0.679962	0.711988	0.697461	0.729071	0.794159
Proportion of births attended by skilled health personnel (%)	0.017481	-0.679962	1.000000	-0.608203	-0.600153	-0.617497	-0.667133
Infant mortality rate (deaths per 1,000 live births):::BOTHSEX	-0.258015	0.711988	-0.608203	1.000000	0.999384	0.999053	0.988302

(Figure 2: Correlation between Infant mortality rate and Under-five mortality rate is 0.988)

Also, normality is one of the key assumptions of a linear regression model. To check the normality of the predictors, I use the QQ-plot. Based on the visualization, it is clear that a few

attributes such as the Adolescent birth rate and the Under-five mortality rate might not fit the normality well. Thus, these attributes might lead to the inaccuracy of the final model.



(Figure 3:QQ-plot of five attributes)

Pros and Cons of the Model

Linear regression model is widely used to determine the association between a dependent variable and other independent variables. It is a straightforward model, so it is easy to interpret.

However, it cannot be applied to the categorical variables. It also does not perform well when the data do not have non-linear relationships.

Unikey:zzha8989 SID:520066234 Executive Summary

My model of choice is linear. For building data models to predict 'Infant mortality rate' for both sexes. The reason for choosing this model is that the linear regression model is straightforward to understand, and the results have good interpretability, which is conducive to decision analysis. And it contains many essential ideas in machine learning. I analyse the correlation with the Infant mortality rate. I picked five attributes.

Code explanation

First, I imported the python that I needed first.

```
from dataclasses import dataclass
import pandas
import numpy
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score

data = pandas.read_csv("SDG_goal3_clean.csv")
data
```

✓ 0.7s Python

	Country	Year	Region	Maternal mortality ratio	Proportion of births attended by skilled health personnel (%)	Infant mortality rate (deaths per 1,000 live births):::BOTHSEX	Infant mortality rate (deaths per 1,000 live births):::MALE	Infant mortality rate (deaths per 1,000 live births):::FEMALE	Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX	Under-five mortality rate, by sex (deaths per 1,000 live births):::MALE	...	Suicide mortality rate, by sex (deaths per 100,000 population):::MALE	Suicide mortality rate, by sex (deaths per 100,000 population):::FEMALE	Death rate due to road traffic injuries, by sex (per 100,000 population):::BOTHSEX	Des i popu
0	Albania	2000	Europe	23	99.1	24.1	27.4	20.6	27.2	30.1	...	7.0	2.8	14.3	
1	Armenia	2000	Asia	43	96.8	27.0	29.8	24.1	30.7	33.8	...	5.1	1.9	19.6	
2	Armenia	2005	Asia	35	97.8	21.3	23.5	18.8	23.9	26.4	...	6.6	2.1	18.3	
3	Armenia	2010	Asia	32	99.5	16.5	18.3	14.6	18.5	20.5	...	10.5	3.5	18.0	
4	Australia	2000	Oceania	7	99.3	5.1	5.6	4.6	6.2	6.8	...	19.9	5.6	9.9	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
158	Ukraine	2005	Europe	33	99.8	12.5	13.9	11.0	14.5	16.0	...	61.9	9.5	25.2	
159	Ukraine	2010	Europe	25	99.9	10.1	11.2	8.9	11.7	12.9	...	44.0	7.7	15.4	
160	Uzbekistan	2000	Asia	41	94.9	51.8	58.6	44.7	62.0	69.3	...	16.0	4.3	9.7	
161	Uzbekistan	2005	Asia	38	100.0	40.4	45.8	34.7	47.1	53.0	...	13.9	4.3	12.9	
162	Uzbekistan	2010	Asia	31	100.0	29.2	33.2	25.0	33.3	37.7	...	11.7	4.5	11.3	

163 rows × 28 columns

This database has no missing data and is a clean dataset. This database has no missing data and is a clean dataset. This dataset has 163 rows and 28 columns. This dataset is about the mortality rate for each country or region. There are twenty-four different causes of death.

```
data = data[['Country','Year','Region','Proportion of births attended by skilled health personnel (%)',
'Infant mortality rate (deaths per 1,000 live births):::BOTHSEX','Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX',
'Neonatal mortality rate (deaths per 1,000 live births)']]
```

✓ 0.3s

I extracted the six required columns I selected. They are 'Country', 'Year', 'Region', 'Proportion of births attended by skilled health personnel (%)', 'Infant mortality rate (deaths per 1,000 live births):::BOTHSEX', 'Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX', 'Neonatal mortality rate (deaths per 1,000 live births)'.

```
data.info()
```

✓ 0.3s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 163 entries, 0 to 162
Data columns (total 7 columns):
 #   Column                                                          Non-Null Count  Dtype
---  ------                                                          --------------  -----
 0   Country                                                         163 non-null    object
 1   Year                                                            163 non-null    int64
 2   Region                                                          163 non-null    object
 3   Proportion of births attended by skilled health personnel (%)   163 non-null    float64
 4   Infant mortality rate (deaths per 1,000 live births):::BOTHSEX  163 non-null    float64
 5   Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX  163 non-null    float64
 6   Neonatal mortality rate (deaths per 1,000 live births)          163 non-null    float64
dtypes: float64(4), int64(1), object(2)
memory usage: 9.0+ KB
```

Then I used data.info() to analyze the type of data. If I want to use. corr() to analyze data correlation, I have to use numerical data for correlation analysis. So ‘Country’ and ‘Region’ do not meet the requirements.

Then I used corr() to analyze data correlation. 'Under-five mortality rate' and 'Neonatal mortality rate' were found to have robust data correlations with our choice of 'Infant mortality rate'.

```
correlations = data.corr()
correlations_target = abs(correlations['Infant mortality rate (deaths per 1,000 live births):::BOTHSEX'])
Relation_percent = correlations_target[correlations_target>0.4]
Relation_percent
```

✓ 0.3s

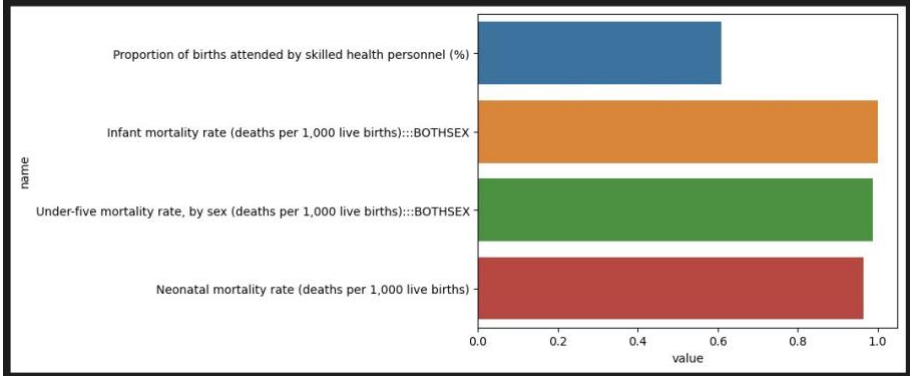
```
C:\Users\user\AppData\Local\Temp\ipykernel_31216\1912283414.py:1: FutureWarning: The default value of numeric_
Select only valid columns or specify the value of numeric_only to silence this warning.
  correlations = data.corr()

Proportion of births attended by skilled health personnel (%)                0.608203
Infant mortality rate (deaths per 1,000 live births):::BOTHSEX               1.000000
Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX   0.988302
Neonatal mortality rate (deaths per 1,000 live births)                       0.964196
Name: Infant mortality rate (deaths per 1,000 live births):::BOTHSEX, dtype: float64
```

```
import matplotlib.pyplot as plt
import seaborn as sbn
df = pandas.DataFrame({'name' :['Proportion of births attended by skilled health personnel (%)',
                                'Infant mortality rate (deaths per 1,000 live births):::BOTHSEX',
                                'Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX',
                                'Neonatal mortality rate (deaths per 1,000 live births)'],
                       'value' :[Relation_percent['Proportion of births attended by skilled health personnel (%)'],
                                 Relation_percent['Infant mortality rate (deaths per 1,000 live births):::BOTHSEX'],
                                 Relation_percent['Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX'],
                                 Relation_percent['Neonatal mortality rate (deaths per 1,000 live births)']]})
sbn.barplot(x = "value",y = "name",data = df)
```

✓ 0.1s

```
<AxesSubplot: xlabel='value', ylabel='name'>
```



I renamed the data name because the data name is too long.

```
data = data.rename(columns={'Proportion of births attended by skilled health personnel (%)':'Proportion of births',
'Infant mortality rate (deaths per 1,000 live births):::BOTHSEX':'Infant mortality rate bothsex',
'Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX':'Under-five mortality rate bothsex',
'Neonatal mortality rate (deaths per 1,000 live births)':'Neonatal mortality rate'})
data
```

✓ 0.3s

	Country	Year	Region	Proportion of births	Infant mortality rate bothsex	Under-five mortality rate bothsex	Neonatal mortality rate
0	Albania	2000	Europe	99.1	24.1	27.2	12.2
1	Armenia	2000	Asia	96.8	27.0	30.7	16.4
2	Armenia	2005	Asia	97.8	21.3	23.9	13.0
3	Armenia	2010	Asia	99.5	16.5	18.5	10.1
4	Australia	2000	Oceania	99.3	5.1	6.2	3.5
...	...	...	...	...	...	...	...
158	Ukraine	2005	Europe	99.8	12.5	14.5	8.7
159	Ukraine	2010	Europe	99.9	10.1	11.7	7.0
160	Uzbekistan	2000	Asia	94.9	51.8	62.0	28.1
161	Uzbekistan	2005	Asia	100.0	40.4	47.1	23.4
162	Uzbekistan	2010	Asia	100.0	29.2	33.3	18.4

163 rows × 7 columns

I used a linear regression model.

```
columns_list = ['Year','Proportion of births','Under-five mortality rate bothsex','Neonatal mortality rate']
```

✓ 0.4s

+ 代码 + Markdown

```
from sklearn.model_selection import train_test_split, GridSearchCV
xtrain, xtest, ytrain, ytest, = train_test_split(data[columns_list],data['Infant mortality rate bothsex'], test_size=0.3, random_state=42)
```

✓ 0.4s

```
from sklearn.linear_model import LinearRegression
LinearR=LinearRegression()
LinearR.fit(xtrain,ytrain)
```

✓ 0.3s

```
def RMSE(actual,prediction):
    return numpy.sqrt(mean_squared_error(actual,prediction))
print('Train RMSE: ', RMSE(LinearR.predict(xtrain),ytrain))
print('Train R-squared: ', r2_score(LinearR.predict(xtrain),ytrain))
print('Test RMSE: ', RMSE(LinearR.predict(xtest),ytest))
print('Test R-squared: ', r2_score(LinearR.predict(xtest),ytest))
```

✓ 0.3s

```
Train RMSE:  0.9719649619203955
Train R-squared:  0.995377136859107
Test RMSE:  2.354745108573993
Test R-squared:  0.981901542372834
```

Both train and test have higher RMSE. This indicates that the data error is significant, and the data collected by the database may need to be completed. But the r2-score is enormous. It is close to 1. The data is reliable, so the information itself may have no problem, but the lack of data content causes deviation. **Advantages and disadvantages of linear regression models** Advantage:

The modelling speed is fast, does not require very complex calculations, and still runs very fast in the case of a large amount of data. Understanding and explaining each variable can also be given in terms of coefficients.

Disadvantage:

It needs to fit nonlinear data better. Therefore, it is necessary first to determine whether there is a linear relationship between the variables. It isn't easy to express highly complex data well.

UNIKEY: scao2237
SID: 520400070

The predictive analysis of infant mortality rate

Dataset description

This report aims to predict the infant mortality rate based on the maternal mortality ratio and neonatal mortality rate. I build two prediction models, the linear regression, and the K-Nearest Neighbors regression. The original dataset is downloaded from the modules of the unit of DATA1002.

There are three columns of data that I used for the predictive analysis.

- Maternal mortality ratio
- Infant mortality rate (deaths per 1,000 live births) BOTHSEX
- Neonatal mortality rate (deaths per 1,000 live births)

Predictive analysis

Build the model---Linear Regression

Firstly, I begin the program by importing all the libraries that I am going to use in the following analysis for the prediction model and read the CSV file into a pandas Data Frame.

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

# Read the csv file
data = pd.read_csv(r"C:\Users\HP\OneDrive\桌面\DATA1002\SDG_goal3_clean.csv")
print(list(data))
# Rename two columns of data that I will use in the following analysis
data.rename(columns={'Infant mortality rate (deaths per 1,000 live births):::BOTHSEX':'Infant death rate'},inplace=True)
data.rename(columns={'Neonatal mortality rate (deaths per 1,000 live births)':'Neonatal mortality rate'},inplace=True)
print(list(data))
```

Then I slice the independent variables which are the maternal mortality ratio and neonatal mortality rate and the dependent variable which is the infant mortality rate and store them in x and y respectively to build the model.

```
# slice dataFrame for input(independent) variables
x = data[['Maternal mortality ratio','Neonatal mortality rate']]
# slice dataFrame for target(dependent) variable
y = data['Infant death rate']
```

Next, I allocate 10% for testing the accuracy of the model later and set a seed to the random process. I used the train test split function to help split the data for training and testing. And finally, create a linear regression model by filling in data.

```
# Split data for training and testing
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=42)
# Build the model with linear regression
regression = linear_model.LinearRegression().fit(x_train, y_train)
```

Predicting target variable with model

By creating a sample to predict the rate of infant date and calling a method that is *predict* on the model, I get the predictive value of the ratio of infant mortality.

```
# Create one example and predict the percentage of infant mortality ratio
sample = [30, 4.5]
sample_predict = regression.predict([sample])
print('----- Sample case -----')
print("Maternal mortality ratio:", sample[0])
print("Neonatal mortality rate:", sample[1])
print('Predicted ratio of infant mortality:', int(sample_predict))
print('----------------------')
```

Output:

```
----- Sample case -----
Maternal mortality ratio: 30
Neonatal mortality rate: 4.5
Predicted ratio of infant mortality: 7
----------------------
```

Testing the accuracy of the model

```
# The coefficients
print('Coefficients:')
print(regression.coef_)
```

```
Coefficients:
[0.02801366 1.6467332 ]
```

From the coefficients, it is obvious that the second element is much greater than the first element, which means that the neonatal mortality rate is associated with the infant mortality ratio more than the maternal mortality rate.

```
# Use the model to predict y from X_test
y_predict = regression.predict(x_test)
# Root mean squared error
mse = metrics.mean_squared_error(y_test, y_predict)
print('Root mean squared error (RMSE):', sqrt(mse))
# R-squared score: 1 is perfect prediction
print('R-squared score:', metrics.r2_score(y_test, y_predict))
```

output:

```
Root mean squared error (RMSE): 4.696816212670311
R-squared score: 0.8934513175887182
```

The output shows that the root means the squared error is approximately 4.697 and the r-squared score is about 0.89. As the interpretation of r-squared is how well the regression model explains observed data, the r-squared I get from above which is 89% determines 89% of the variance in the dependent variable (infant mortality rate) is explained by the independent variables (the maternal mortality ratio and neonatal mortality rate), which means that these data fit the regression model well.

Method 2 Build the model---K-Nearest Neighbors Regression

```
import pandas as pd
from math import sqrt
from sklearn import metrics
from sklearn import neighbors
from sklearn.model_selection import train_test_split

# Read the csv file
data = pd.read_csv(r"C:\Users\HP\OneDrive\桌面\DATA1002\SDG_goal3_clean.csv")
print(list(data))
# Rename two columns of data that I will use in the following analysis
data.rename(columns={'Infant mortality rate (deaths per 1,000 live births):::BOTHSEX':'Infant death rate'},inplace=True)
data.rename(columns={'Neonatal mortality rate (deaths per 1,000 live births)':'Neonatal mortality rate'},inplace=True)
print(list(data))
```

Using the same dependent variable and independent variables to build the model.

```
# slice dataFrame for input(independent) variables
x = data[['Maternal mortality ratio','Neonatal mortality rate']]
# slice dataFrame for target(dependent) variable
y = data['Infant death rate']
# Split data for training and testing
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=42)
# Build the k_nearest neighbors regression
neigh = neighbors.KNeighborsRegressor(n_neighbors=4).fit(x_train, y_train)
# Create one sample and predict the ratio of infant mortality
sample = [30, 4.5]        # a sample with the maternal mortality ratio is 30 and the neonatal mortality rate is 4.5
sample_predict = neigh.predict([sample])
print('----- Sample case -----')
print("Maternal mortality ratio:",sample[0])
print("Neonatal mortality rate:",sample[1])
print('Predicted ratio of infant mortality:', int(sample_predict))
print('-----------------------')
```

Output:

```
----- Sample case -----
Maternal mortality ratio: 30
Neonatal mortality rate: 4.5
Predicted ratio of infant mortality: 8
-----------------------
```

```
# Use the model to predict X_test
y_predict = neigh.predict(x_test)
# Root mean squared error
mse = metrics.mean_squared_error(y_test, y_predict)
print('Root mean squared error (RMSE):', sqrt(mse))
# R-squared score: 1 is perfect prediction
print('R-squared score:', metrics.r2_score(y_test, y_predict))
```

Output:

```
Root mean squared error (RMSE): 5.190386333801912
R-squared score: 0.8698811189296204
```

The output shows that the R-squared score is approximately 0.87, which indicates that the two independent variables have a strong effect on the dependent variable.

Conclusion

In general, linear regression is a more applicable measure to predict the ratio of infant mortality based on maternal mortality ratio and neonatal mortality rate than K-Nearest Neighbors regression. The lower the root mean squared error, the greater the accuracy of the model, and, on the contrary, the higher the value of the R-squared score, the better the model is, it is obvious that the R-squared error of linear regression is closer to 1 than that is in K-Nearest Neighbors regression, I figure out that the linear regression is a better measure of prediction for such data.

Furthermore, the advantage of linear regression is that it can be easier to interpret the output coefficients and then to see the relationship between independent variables and

dependent variables obviously and clearly. But linear regression is an assumption between dependent variables and independent variables. And for the K-Nearest Neighbors regression, is useful for some cases where the data do not have a linear relationship, but it is quite inefficient in computation and the accuracy of prediction in my case is not well.

Limitation

To achieve a more accurate prediction in the model, this may require more attributes to find the relationship between independent variables and dependent variables than just two attributes.

Summary

Model 1

The K-nearest regression predictive model is used to predict the infant mortality rate. There are two attributes will be used in this prediction as following:

> *Infant mortality rate (deaths per 1,000 live births)::: BOTHSEX* and *Health worker density, by type of occupation (per 10,000 population) ::PHYSICIAN.*

By using the K-nearest regression predictive model, the RMSE is 16.327469575996826 and the R-squared score is 0.21384875954566485. The RMSE is quite higher than we expected and the R-squared score implies that there is a quite weak relationship between these two attributes in terms of this KNN regression model.

Model 2

This model is a Linear regression model with 5 attributes: *Adolescent birth rate, Universal health coverage service coverage index, Maternal mortality ratio, Mortality rate attributed to cardiovascular disease, cancer, diabetes or chronic respiratory disease.* The final model equation is shown as below.

$$Y = 0.7972006 - 0.00557274X_1 - 0.01193058X_2 - 0.2224974X_3 + 0.04871886X_4 + 0.81724752X_5$$

The RMSE of this model is around 1.1457, which implies a good fit of this model. The R-square is around 0.99. Based on the in-sample test, the accuracy of this model is approximately 92.638%.

Model 3

This is a linear regression model. Contains *'Proportion of births attended by skilled health personnel (%)', 'Under-five mortality rate, by sex (deaths per 1,000 live births):::BOTHSEX', 'Neonatal mortality rate (deaths per 1,000 live births)'*. Finally, four sets of data are obtained as Training RMSE: 0.9719649619203955, Training R-squared: 0.995377136859107, Test RMSE: 2.354745108573993, Test R-squared: 0.981901542372834.

The two R-squared numbers are more than 0.2~0.5, indicating that the accuracy of the data prediction could be higher. Incomplete data may cause it. But the two R2-scores are astonishingly above 0.9. It shows that the model fitting degree is relatively high, so the reference value is very high.

Model 4

Using two models which are linear regression and K-Nearest Neighbors regression to predict the infant mortality rate based on the maternal mortality ratio and neonatal mortality rate, it can be find that the R-squared score got from these two models are 0.89 and 0.87 respectively and the RMSE are 4.697 and 5.190 respectively. As the higher the value of the R-squared score, the better the model is, and the lower the RMSE, the greater the accuracy of the model, so linear regression is a more applicable measure to predict the ratio as the data fit the linear regression well and more accurately.

Conclusion

Based on the prediction result, it suggests that Maternal mortality ratio, Universal health coverage service coverage index, and Adolescent birth rate have relatively strong correlation with Infant mortality rate. We can assume that the country with high adolescent birth rate or high maternal mortality ratio tends to have a high infant mortality rate.

From the perspective of machine production, we can see that the Linear regression model tends to have a higher R-squared value than the K-nearest Neighbors regression model based on comparison with these models. In another word, the Linear regression model is more likely to have a higher accuracy than the other one when we need to predict the numerical variables in our analysis.

The pros and cons of KNN regression: The KNN model is a better choice when the data is not just a nearly linear relationship but has quite complex patterns. And KNN model is not disturbed that much than linear regression model by a few outliers or errors. However, when using the KNN model, it tends to take more computation and storage.

The pros and cons of linear regression: Linear regression is good at capturing linear relationships in a dataset. Moreover, the model is updated faster when new data is added, and no parameter adjustment is required. The results are interpretable and easy to interpret. However, it is not suitable for nonlinear data and has limitations. Its prediction accuracy is low, and it can only be used as a reference and not as favourable evidence. The most important point is that linear regression can overfit.

Among the three different Linear regression models, the more input attributes we have, the higher the R-squared value we received. It suggests that with more accurate attributes we put into the predict model, there are more chances we receive the correct prediction result.

Limitation

In the prediction of using KNN model, some outliers come out in the attributes that we are focusing on and it does have an impact on the result. That will influence the whole prediction. Maybe the outliers need to be cleaned before the prediction.

The data 'Under-five mortality rate bothsex' is controversial. Since the data will contain 'Infant mortality rate' data, the data will overlap. Therefore, this data has both a high r2-score and a high RMSE. This is probably why the correlation is as high as 0.98.