MA335 Final Project

Ahmad Afzaal Cheema

Reg. no 2111168

31st March 2022

Abstract

In this report, we need to analyze the relationship between World Development Indicator and Covid-19 death rate. We do clustering using K-means clustering algorithm by which we make 6 clusters, equals to number of continents. We use logistic regression for binary classification to foresee high COVID causalities and get 100% accuracy. For multi class classification we use two more algorithms i.e. Linear Discriminant Analysis and Quadratic discriminant analysis along with Logistic Regression and then compare their results, LDA gives highest accuracy of 89.3% among all three of them.

Contents

Total Word Count: 2209

Introduction

In this project, the dataset which we get from World Bank Database includes different World Development Indicators, on which we have to perform clustering like K-Means clustering as well as classification techniques like Logistic Regression, Linear discriminant analysis (LDA), Quadratic discriminant analysis (QDA) to understand the relation between world Development Indicator and seriousness of COVID-19.

Preliminary Analysis

Pre-processing of Dataset

For high accuracy we need to refine dataset. It contains many NA values. To solve this problem we add some values that are possible like continent name with the help of country name and drop rows that are completely null like last row. We can deal with other NA values for investigation as indicated by need.

The table below shows the summary of dataset:

Column Name	Distinct Rows	Missing Values
Country	185	0
Continent	6	1
Covid_deaths_rate	169	0
Life_expectancy	177	8
Electricity_access	81	0
Net_national_income	133	52
Income_per_capital	133	52
Mortality	134	13
Primary	115	70
Pop_growth	184	1
Pop_density	185	0
Pop_total	184	1
Health_expenditure_pci	167	18
Health_expenditure	167	18
Unemployement	107	73
Gdp_growth	175	10
Gdp_capita	176	9
Birth_rate	159	5
Water_services	102	77
Comp_education	14	12

Table 1: Dataset Summary

As shown in the table every column have some missing values, so we need to do some dataset handling before implementation of any algorithm.

Descriptive Statistics of Columns with numerical Values:

	count	mean	std	min	25%	50%	75%	max
covid_deaths_rate	185.0	1.143378e+03	1.219881e+03	3.000000	1.670000e+02	7.110000e+02	1.830000e+03	6.252000e+03
life_expectancy	177.0	7.301281e+01	7.488963e+00	54.239000	6.727300e+01	7.447500e+01	7.849756e+01	8.507805e+01
electricity_access	185.0	8.573146e+01	2.499595e+01	6.720535	8.350000e+01	1.000000e+02	1.000000e+02	1.000000e+02
net_national_income	133.0	4.280545e+00	6.105832e+00	-14.378665	1.304721e+00	3.653306e+00	6.270332e+00	5.017190e+01
income_per_capita	133.0	2.931443e+00	5.879021e+00	-17.347416	5.296974e-01	2.637275e+00	5.076274e+00	4.725177e+01
mortality	172.0	2.045640e+01	1.938805e+01	1.600000	5.250000e+00	1.305000e+01	3.162500e+01	8.240000e+01
primary	115.0	9.289936e+01	1.296284e+01	54.728691	8.619812e+01	9.740717e+01	1.011704e+02	1.204473e+02
pop_growth	184.0	1.241058e+00	1.124111e+00	-1.609508	4.265082e-01	1.164656e+00	1.987952e+00	4.468688e+00
Pop_density	185.0	3.773194e+02	1.634906e+03	2.071058	3.589318e+01	8.419527e+01	2.170076e+02	1.922398e+04
Pop_total	184.0	4.117526e+07	1.492641e+08	33706.000000	2.116046e+06	9.370390e+06	3.037900e+07	1.407745e+09
health_expenditure_pci	167.0	1.196708e+03	1.915711e+03	19.849976	7.380273e+01	3.701100e+02	1.254541e+03	1.092101e+04
health_expenditure	167.0	6.431587e+00	2.589611e+00	1.525117	4.398296e+00	6.243371e+00	8.102088e+00	1.676706e+01
Unemployment	112.0	7.348571e+00	5.237680e+00	0.100000	3.787500e+00	5.350000e+00	9.952500e+00	2.847000e+01
gdp_growth	175.0	2.918907e+00	3.108137e+00	-7.157247	1.282986e+00	2.653066e+00	4.889622e+00	1.953581e+01
gdp_capita	176.0	1.885117e+04	2.889249e+04	228.213589	2.072459e+03	6.731240e+03	2.334739e+04	1.894871e+05
birth_rate	180.0	1.939355e+01	9.887596e+00	5.900000	1.050000e+01	1.733700e+01	2.713525e+01	4.563700e+01
water_services	108.0	7.360029e+01	2.948601e+01	5.581210	5.538591e+01	8.708354e+01	9.873139e+01	1.000000e+02
comp_education	185.0	3.394595e+00	3.998772e+00	0.000000	0.000000e+00	0.000000e+00	8.000000e+00	9.000000e+00

Figure 1: Statistics of Columns with numerical Values

We also check correlation of data, to check how different variables (key Features) are related to each are they are dependent to each other or Independent in Nature. As shown in the figure 8(in the appendix) we checked that my target feature Covid death rate how much they depends on other features in the dataset. The range of correlated feature are lies between -1 to 1. Here if the value is in between 0 to 1 tell us that the feature are dependent to each other and for independent nature values will below than 0.

Scatter Plot:

We plot a scatter plot to show the relationship between water services and life expectations in every continent and with total population. The size of the circle shows the total population. The graph shows that Africa have the lowest water services. As we move to the right side of the graph it shows that Europe have very good water services which increases the chances of life expectation in Europe.

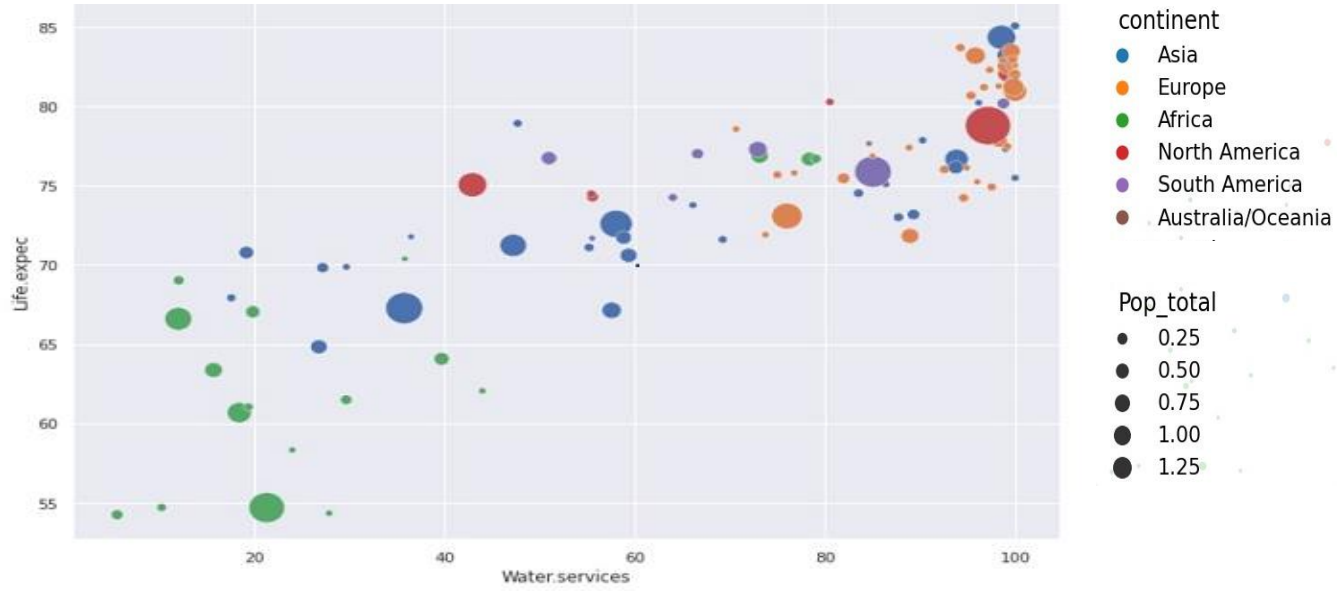

Figure 2: Relationship between Life Expectation and Water services in continents and Total population

Box Plot:

We plot box plot against continents and covid_death_rate. To shows that how much the countries of each continent is affected.

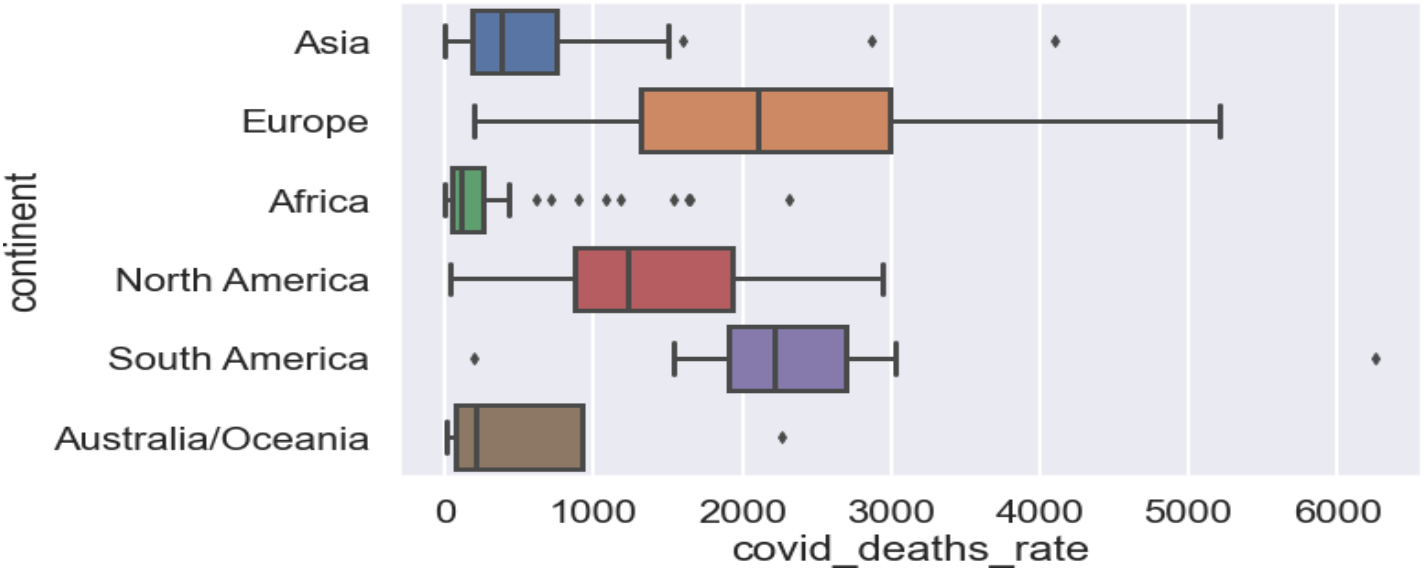

Figure 3: Death rate in Continents

1- Africa: In Africa the box plot tell us have death rate from 104 to 1580

2- Asia: In Asia the box plot tell us have death rate of different countries in Asia is from 177 to 1580

3- Europe: In Europe the death rate is normally distributed the death rate of countries lies between 1315 to 4539 the fifty percent of the countries in Europe has death rate from 1315 to 2103 and the other 50 percent is 2103 to 4539

4- North America: In North America the death rate of most of countries lies between 1228 to 2564

5- South America: In North America the death rate of most of countries lies between 2210 to 4477

6- Australia/Oceania: In North America the death rate of most of countries lies between 206 to 985

Analysis:

Clustering:

We divide the dataset into certain groups known as clusters which have some general similar characteristics. We use K-means clustering algorithm for this purpose.

In this algorithm we first assign number of clusters in our case its k=5. Then randomly assign data point to each clusters. Afterwards we calculate centroids for each cluster and then re-assign data points to each cluster which is close to the centroid of the cluster. Repeat the cycle of calculating centroid and re-assignment of data points.

Optimal Selection of number of clusters:

We use Silhouette Score for the adequate selection of the number of clusters. Silhouette Score is one of the most popular method to calculate the effectiveness of clusters. It ranges from -1 to 1, where -1 represents the wrong assignment whereas 1 shows that cluster are well separated and recognized.

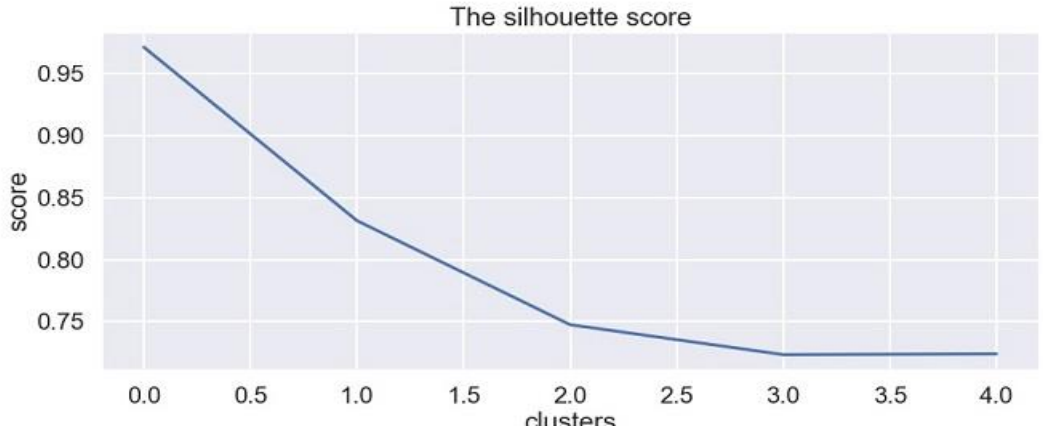

Figure 4: Silhouette Score

After using Silhouette Score we update the number of cluster from 5 to 6 clusters (k=6), as we are getting high score after k=6. Data become distributed among 6 clusters which have close centroids. The resulting clusters shows that we have number of clusters we have are equal to the number of continents which means that we can use continents as clusters to divide the data in clusters

Binary Classification using Logistic Regression:

Logistic Regression is supervised learning classification algorithm which is used to foresee the likelihood of an objective variable.

In binary classification of logistic regression there are only two classes 0 and 1.

We train the model by using all variables like water services, life expectation, GDP growth, GDP capital, income net capital, health expenditure, birth rate, unemployment rate etc. as independent variables and covid_death_rate as dependent variable.

After training the model we get 100% accuracy for binary classification. Figure 5 shows confusion matrix for binary classification.

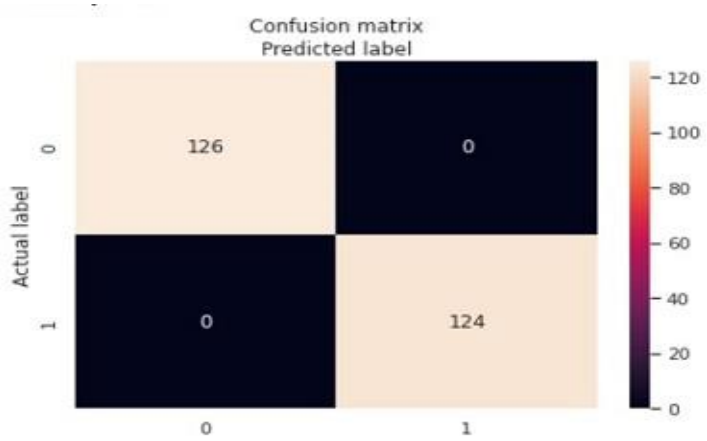

Figure 5: Confusion matrix of Binary Classification

Classification using QDA, LDA and Logistic Regression:

Linear discriminant analysis (LDA) and Quadratic discriminant analysis (QDA) both are also classification techniques as logistic Regression.

LDA is used for linear classification whereas QDA is a variable of LDA that takes into account for non-linear classification.

Data pre-processing:

Before performing the multi class classification first we need to remove all the Na values from the data to calculate the min max.

Once we remove the null values and calculate min max, it is ready to perform multi class classification.

As we have to do multi-class classification we first create four classes namely low, med, high and very high in corresponded to COVID death rate.

The table below show the range of Covid deaths for each class.

Class Name	Range
Low	0-800
Med	800-1600
High	1600-2400
Very high	2400-8000

Table 2: Covid_death_rate Class Ranges

Result comparison:

We apply all the three algorithms on the dataset and compute following results:

Algorithm	Accuracy	Precision	Recall
LDA	0.893	0.893	0.893
QDA	0.769	0.769	0.769
LR	0.617	0.617	0.617

Table 3: Result Comparison of multi-class Classification

The results shows that out of all the three classification algorithms LDA performs very well and give the highest accuracy among all of them. Whereas Logistic regression does not perform very well in multi class classification it gives only 61.7% accuracy which is lowest from all of them

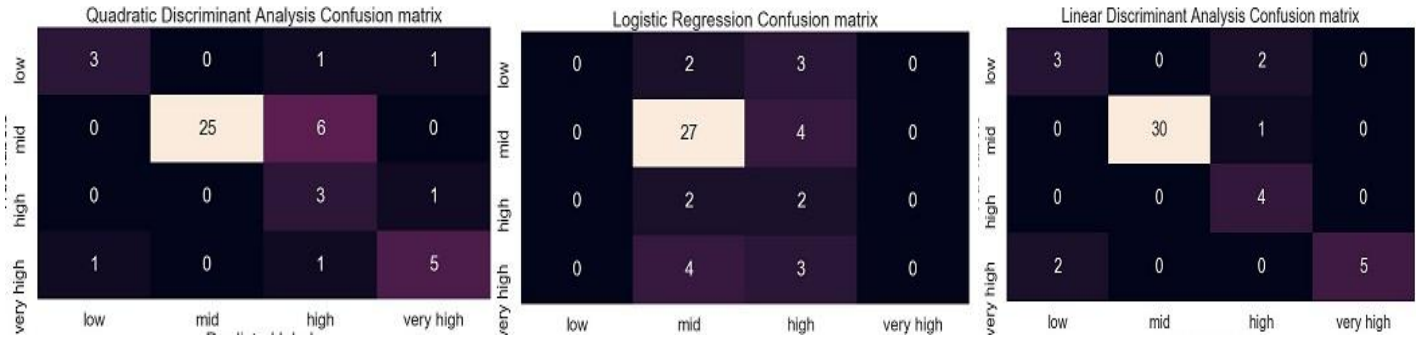

Figure 6: Confusion matrices of QDA, LR and LDA for Multi classes

Discussions (Task 5):

After analyzing the dataset , we observed that south- America is most affected continent by COVID-19 as shown in the figure 7.We also see that indicators like water services , total population, life expectation plays an important to predict that which one of the continent is most affected and is on great risk of coronavirus spread. These indicators have covered many important aspects that may play a role in spread or ceasing of the pandemic like indicators water services shows that where water services are better there are more expectation of life or vice versa. Same in case of health_expenditure, unemployment, national income etc. also have very important role, more resources the continent or country have more they have control over the pandemic.

This shows that these World Development Indicators can predict causalities of any similar pandemic to great extent.

We have seen that countries which have good national income, good GDP growth and less unemployment rate have better economic condition which in return have great chances to better and rapidly deal with the pandemic whereas countries have not very good economic profile may faces problem to deal with the pandemic.

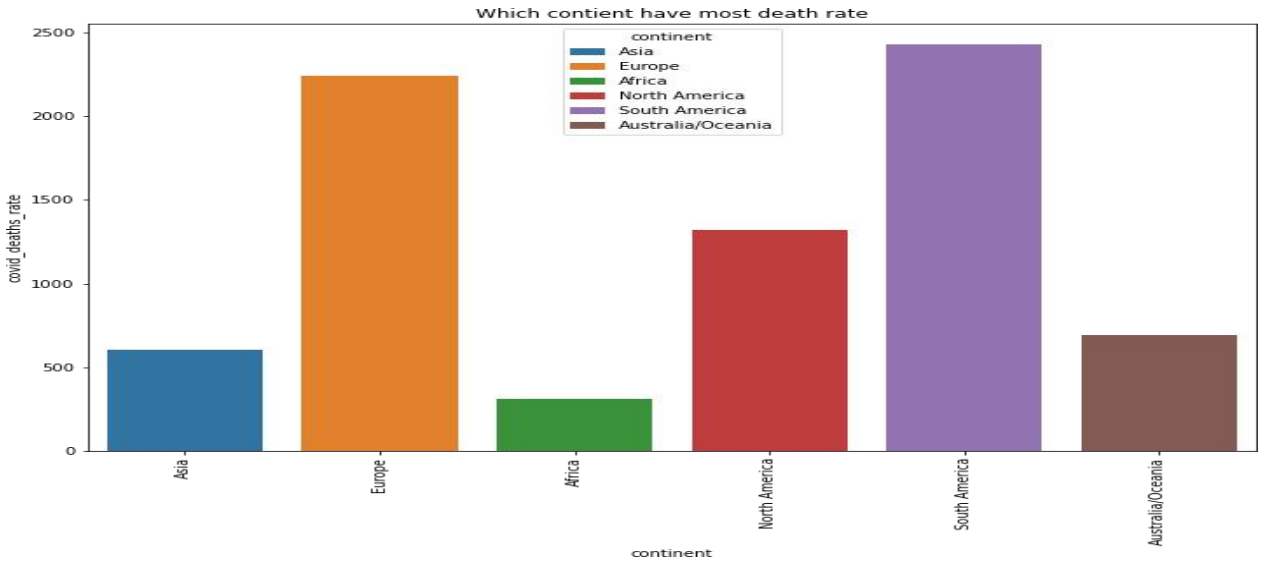

Figure 7: Covid death rate in continents

Conclusions:

World Development Indicators and Covid_19 have close relationship. After analyzing the data we can see that South America is most affected continent by covid-19. We also see that Silhouette Score plays an important role in optimal selection of clusters during clustering (that are k=6, equals to the number of continents) which makes results more effective.

Logistic regression is used for both binary as well as for multi-class classification. It gives 100% accuracy in 2 classes whereas in multi class it only produces 61.7% which means it came down on accuracy graph as we move from binary to multiple classes.

However Linear Discriminant Analysis performs very well in multi class classification it out-performs both Logistic regression as well as Quadratic discriminant analysis and gives highest accuracy of 89.3%

References:

An Introduction to Clustering and different methods of clustering sauravkaushik8 https://www.r-bloggers.com/2020/11/lda-vs-qda-vs-logistic-regression/

Silhouette Coefficient Ashutosh Bhardwaj

Appendix:

- **Import all necessary Libraries:**

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

from sklearn.metrics import accuracy_score

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis,QuadraticDiscriminantAnalysis

import warnings

import io

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import silhouette_score

%matplotlib inline

sns.set()
```

- **Loading Dataset:**

```
data = pd.read_csv('project.csv')
data['Covid.deaths']=pd.to_numeric(data['Covid.deaths'].str.replace(',',''))
data['Comp.education']=pd.to_numeric(data['Comp.education'].str.replace('..','0'))
data=data.replace(np.nan,0)
Project_data = data.copy()
Project_data.head()
```

- **Checking Correlation:**

```
cor = Project_data.corr()
plt.figure(figsize=(15,8))
sns.heatmap(cor,annot=True)
```

- **Describing Dataset:**

```
Project_data.describe().T
```

- **I find an Insight and then I draw it:**

```
plt.figure(figsize=(12,8))
plt.title('Life expactancy in different Continents according to the water resourser')
sns.scatterplot(x='Water.services',y='Life.expec',hue='Continent',data=Project_data,size='Pop.total',legend=False,s
izes=(20, 4000))
plt.show()
```

- **Creating Histograms:**

```
sns.set_context('poster')
plt.figure(figsize=(35,20))
for i in range(3,19):
  plt.subplot(3,6,i)
  plt.title(Project_data.columns[i])
  plt.hist(Project_data.iloc[:,i])
  plt.show()
```

- **Creating boxplot:**

```
plt.title("Box plot continent againast covid death rate\n")
boxplot = Project_data.boxplot(figsize = (4,19), rot = 90, fontsize= '5', grid = False)
plt.figure(figsize=(2,1))
sns.boxplot(y='Continent', x='Covid.deaths', data=Project_data, orient='h')
plt.show()
```

- **Data preprocessing:**

```
from sklearn.preprocessing import MinMaxScaler
print(Project_data)
X = Project_data.iloc[:,3:]
df = X.dropna()
scaler = MinMaxScaler()
new=scaler.fit_transform(df)
scaled=pd.DataFrame(columns=df.columns,data=new)
```

- **Clustering:**

```
kmeans = KMeans(n_clusters=5, random_state=0)
kmeans.fit(scaled)
pred = scaled.copy()
pred['kmean1'] = kmeans.labels_
pred.head()
```

- **For Optimal Numbers of Clusters:**

```
sil = []
kmax = 6
```

- **Dissimilarity would not be defined for a single cluster, thus, minimum number of clusters should be 2**

```
for k in range(2, kmax+1):
   kmeans = KMeans(n_clusters = k).fit(df)
   labels = kmeans.labels_
   sil.append(silhouette_score(df, labels, metric = 'euclidean'))
plt.figure(figsize=(15,6))
plt.plot(sil)
sns.set_context('poster')
plt.xlabel('clusters')
plt.ylabel('score')
plt.title('The silhouette score')
plt.show()
Project_data = Project_data.dropna()
```

- **Binary Classification With Logistic Regression**

```
X, y = make_blobs(n_samples=1000, centers=2, random_state=1)
```

- **summarize observations by class label**

```
count = Counter(y)
```

- **plot the dataset and color the by class label**

```
for i in range(10):
```

```
for label, _ in count.items():
  row_ix = where(y == label)[0]
  plt.scatter(X[row_ix, 0], X[row_ix, 1], label=str(label))
plt.legend()
plt.show()
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25)
```

- **Create an instance of the model.**

```
logregbinary = LogisticRegression()
```

- **Training the model.**

```
logregbinary.fit(X_train,y_train)
```

- **Do prediction.**

```
y_pred=logregbinary.predict(X_test)
```

- **Analyzing the results**

```
from sklearn import metrics
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
cnf_matrix
```

- **name of classes**

```
class_names=[0,1]

fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
```

- **create heatmap**

```
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

- **Applying Logistic for multi classes:**

```
from sklearn.metrics import classification_report, confusion_matrix
feature_cols  =['Covid.deaths','Life.expec','Elect.access','Net.nat.income','Net.nat.income.capita',
        'Mortality.rate','Primary','Pop.growth','Pop.density','Pop.total','Health.exp.capita','Health.exp',
        'Unemployment','GDP.growth','GDP.capita','Birth.rate','Water.services','Comp.education']
X = Project_data[feature_cols].replace(",", "", regex=True) # Features
y = Project_data.iloc[:,2].replace(",", "", regex=True)
yy = pd.DataFrame(y)
y=pd.cut(yy['Covid.deaths'], bins=[0,  800, 1600, 2400,8000], include_lowest=True, labels=['low', 'mid', 'high','ve
ry high'])
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=1)
```

- **instantiate the model (using the default parameters)**

```
logreg = LogisticRegression()
```

- **fit the model with data**

```
logreg.fit(X_train,y_train)
y_pred=logreg.predict(X_test)
from sklearn import metrics
cm = confusion_matrix(y_test, y_pred, labels=logreg.classes_)
plt.figure(figsize=(15,6))
ax= plt.subplot()
sns.heatmap(cm, annot=True, fmt='g', ax=ax);
```

- **labels, title and ticks**

```
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Logistic Regression Confusion matrix');
ax.xaxis.set_ticklabels(y.unique()); ax.yaxis.set_ticklabels(y.unique());
plt.show()
print("\n\nLogistic Regression")
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
print("Precision:",metrics.precision_score(y_test, y_pred , average='micro'))
print("Recall:",metrics.recall_score(y_test, y_pred ,average='micro'))
```

- **Applying QDA:**

```
qda=QuadraticDiscriminantAnalysis(reg_param=0.95)
qda.fit(X_train,y_train)
pred_qda=qda.predict(X_test)

cm = confusion_matrix(y_test, pred_qda, labels=logreg.classes_)
plt.figure(figsize=(15,6))
ax= plt.subplot()
sns.heatmap(cm, annot=True, fmt='g', ax=ax);
```

- **labels, title and ticks**

```
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Quadratic Discriminant Analysis Confusion matrix');
ax.xaxis.set_ticklabels(y.unique()); ax.yaxis.set_ticklabels(y.unique());
plt.show()
print("\n\nQuadratic Discriminant Analysis")
print("Accuracy:",metrics.accuracy_score(y_test, pred_qda))
print("Precision:",metrics.precision_score(y_test, pred_qda , average='micro'))
print("Recall:",metrics.recall_score(y_test, pred_qda ,average='micro'))
```

- **Applying LDA:**

```
lda = LinearDiscriminantAnalysis()
lda = lda.fit(X_train, y_train)
pred_lda=lda.predict(X_test)
cm = confusion_matrix(y_test, pred_lda, labels=logreg.classes_)
plt.figure(figsize=(15,6))
ax= plt.subplot()
sns.heatmap(cm, annot=True, fmt='g', ax=ax);
```

- **labels, title and ticks**

```
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
ax.set_title('Linear Discriminant Analysis Confusion matrix');
ax.xaxis.set_ticklabels(y.unique()); ax.yaxis.set_ticklabels(y.unique());
plt.show()
print("\n\nLinear Discriminant Analysis")
print("Accuracy:",metrics.accuracy_score(y_test, pred_lda))
```

print("Precision:",metrics.precision_score(y_test, pred_lda , average='micro'))

print("Recall:",metrics.recall_score(y_test, pred_lda ,average='micro'))

Data Correlation:

Figure 8:Data Correlation

Histograms:

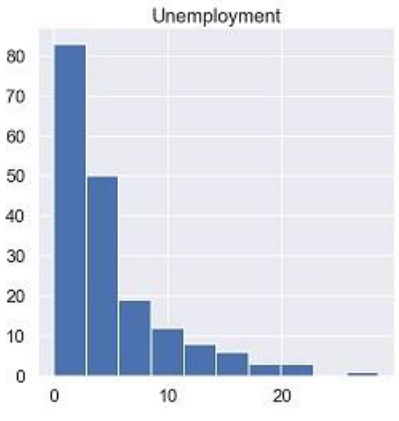

Figure 9 Unemployment: Histogram

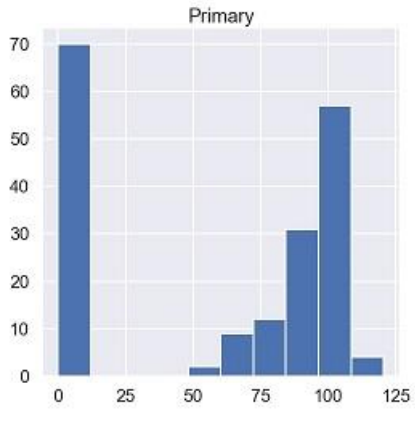

Figure 10: Primary Histogram

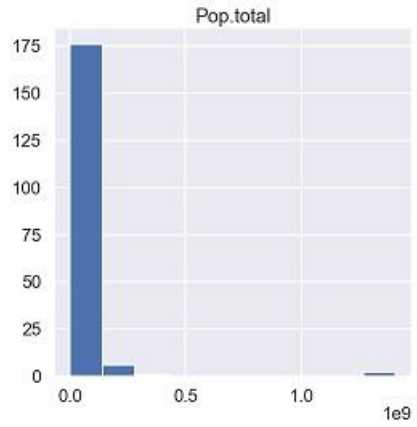

Figure 11: Total Population Histogram

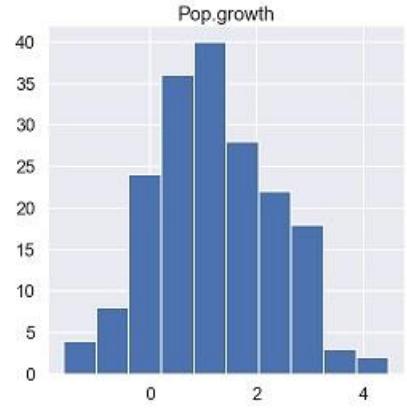

Figure 12: Population growth Histogram

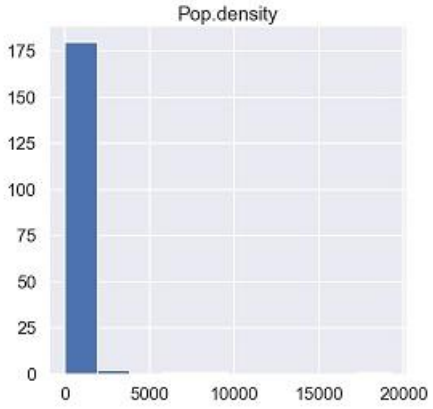

Figure 13: population density Histogram

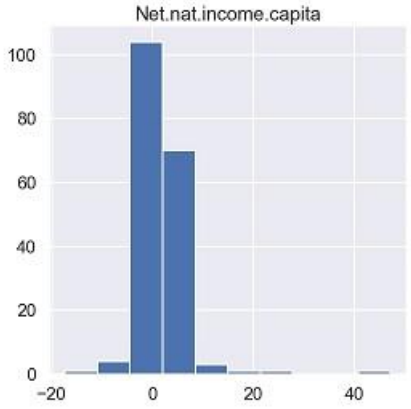

Figure 14: Net nat income capita Histogram

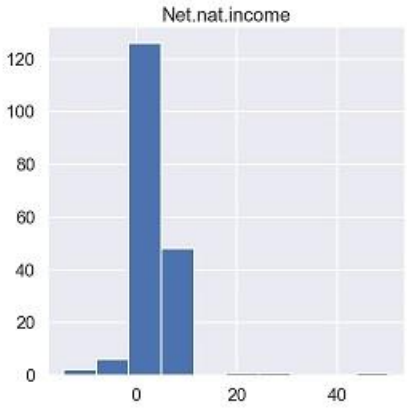

Figure 15: Net nat income Histogram

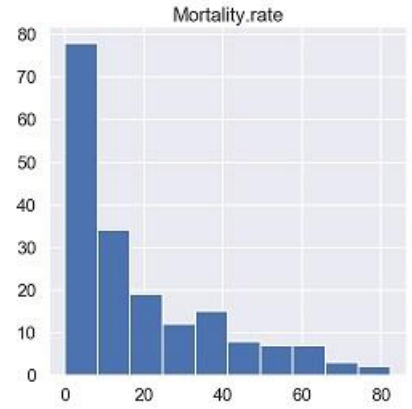

Figure 16: Mortality rate Histogram

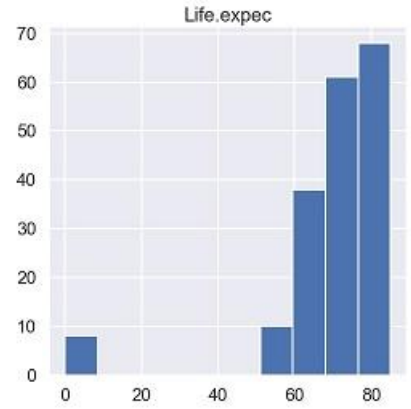

Figure 17: Life expectation Histogram

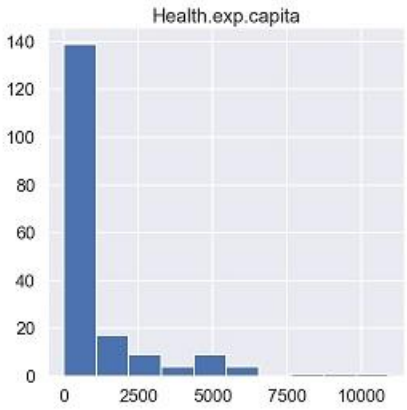

Figure 18: Health expenditure capita Histogram

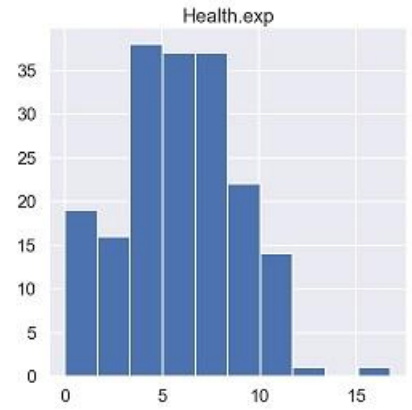

Figure 19: Health Expenditure Histogram

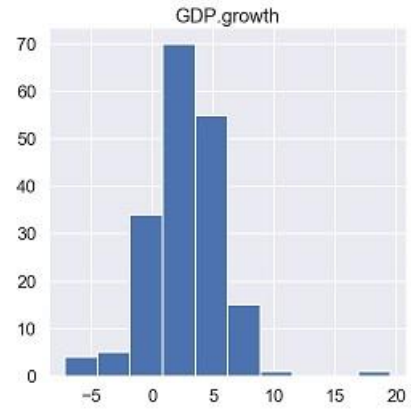

Figure 20: GDP growth Histogram

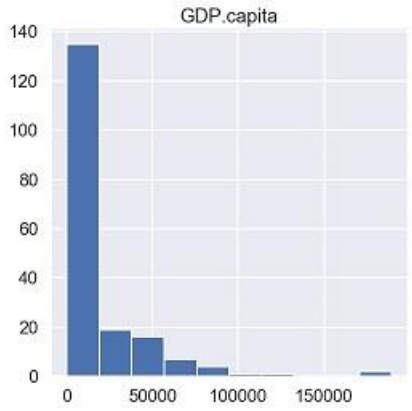

Figure 21: GDp capita Histogram

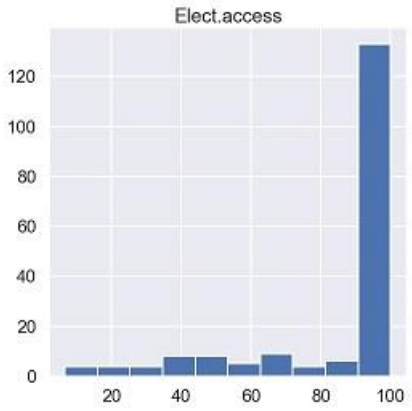

Figure 22: Electric Access Histogram

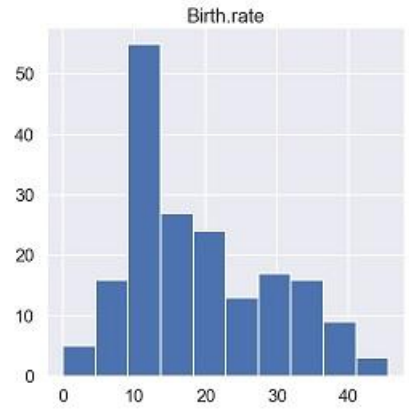

Figure 23: Birth rate Histogram

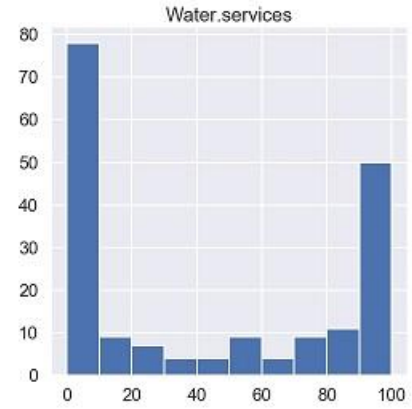

Figure 24: Water Services Histogram