COMP615 – Foundations of Data Science

Lab 4 – Correlation, Feature Selection and Regressions

Introduction

This lab will cover some of the feature selection and simple linear regression method supported by Python. All of the methods discussed in the lectures will form part of the lab. Use the sklearn documentation online to configure the methods. Above all, make sure you understand what you are doing; simply configuring the methods is not enough without an understanding of how they work. The basic code appears below.

Submit: To Do 1:3

1. Case Study: Multidimensional Poverty Measures

Multidimensional poverty measures can be used to create a more comprehensive of poverty. Poverty has a much broader meaning than lack of money. Poverty causes multiple disadvantages at the same time such as malnutrition, a lack of electricity, poor quality of work or schooling. Multidimensional poverty measures reveal which countries are poor and how this poverty affected the life of their nation. Multidimensional measures can be broken down to reveal the poverty level in different areas of a country, and among different sub-groups of people.
In this case study you will explore these factors. To start, download and save the Poverty_LifeExp.csv file under the lab material.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# scaling
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

# linear regression
from sklearn import linear_model
```

```
Pov_data = pd.read_csv("Poverty_LifeExp.csv")
# quick view of columns and values
Pov_data.head()
# how many columns and rows in dataframe
Pov_data.shape
Pov_data.isnull().sum()
# are there duplicate values?
format(len(Pov_data[Pov_data.duplicated()]))
# standard statistical measures
Pov_data.describe(percentiles = [.25, .5, .75, .90 ,.95, .99])
```

1.1. Visualisation

Use histogram to explore distribution of your data.

```
plt.figure(figsize=(12,5))
plt.title("Child Mortality: Death of children under 5 years of age per
1000 live births")
ax = sns.histplot(Pov_data["child_mort"])
```

Perform the above for remaining columns and explain your observation.

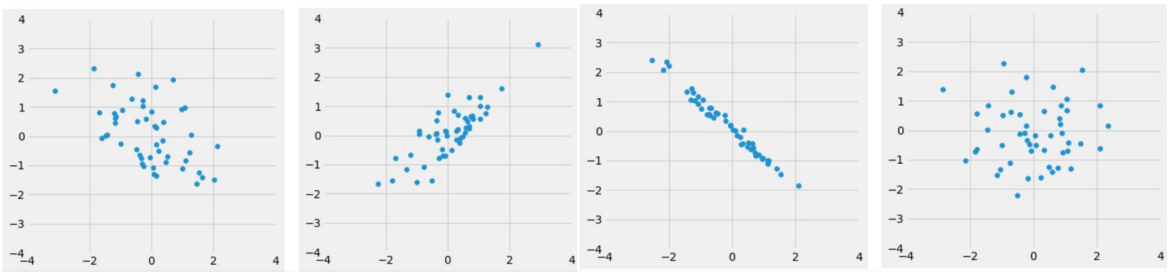
Looking at the data distribution we can see that there are some features that do indeed have outliers. For the purpose of this analysis, outliers will not be removed since they could be considered very informative in that they could point out countries that are in critical condition and in need of help. For example, Child Mortality is a strong indicator of poverty and necessity, so the outliers in this feature show that there are countries with a higher than normal/critical number in child mortality.

1.2. Correlation Analysis

An important aspect of data science is using data to make *predictions* about the future, using information that we currently possess. A question one might ask would be "Given the US GDP of every year of the previous decade, how can we predict the US GDP for next year?" In order to answer this question, we will investigate a method of using one variable to predict another by looking at the *correlation* between two variables.

Look at the following four datasets. Rank them from weakest correlation to strongest correlation.

A B C D



```
# pearson
plt.figure(figsize=(15,10))
sns.heatmap(Pov_data.corr(method='pearson', min_periods=1),annot=True)
```

```
Pov_data.corr()

pd.plotting.scatter_matrix(Pov_data,figsize=[20,20])
plt.show()
```

To Do 1: Provide the results of above correlation analysis and explain your findings.

1.3. Scaling

The features have incomparable units (metrics are percentages, dollar values, whole numbers) the range values of the features also vary (one for example is 0 to 200, and another 0 to 100,000), so here for example, a change of 50 in one feature is quite significant, whereas in another it is almost unnoticeable this level of variance can negatively impact the performance of this model, as this model is based on measuring distances, it can do this by giving more weight to some features by scaling we are removing potential bias that the model can have towards features with higher magnitudes. Scaling only applies to numeric values. Review your dataset and drop the column(s) with values other than integer.

```
Pov_data_Drop = Pov_data.drop([columnname], axis =1) # eliminate the column. Save the new dataset as Pov_data_Drop so you have a backup of original dataset just in case!
Pov_data_Drop.head()
```

There are different methods of scaling data. MinMaxScaler scales data by normalising it.

```
# Columns argument ==> we'll use this later to create a new dataframe
 # with the rescaled data
columns = Pov_data_Drop.columns

scaler = MinMaxScaler() # for the rescaling

#'fit' function is to find the x_min and the x_max
#'transform' function applies formula to all elements of data

normalised_dataset = scaler.fit_transform(Pov_data_Drop)

normalised_dataset

My_ normalised_df = pd.DataFrame(data normalised_dataset , columns =
columns )
My_ normalised_df
```

Study StandardScaler (standardised) and apply it on 'Pov_data_Drop' dataset. Provide the code and observe the results.

2. Linear Regression Model

Download the 'MPI_Dataset.csv' dataset. A new feature called *Multidimensional Poverty Index (MPI)*, international measure of acute multidimensional poverty covering over 100 developing countries, is included in this dataset. In this exercise MPI is the dependent variable (to be predicted) and the data we already been working on will act as the independent variables (predictors).
See appendix A (metadata)

```
# Perform step 1:3 first:

# 1) Import data and save it as 'mpi_ds'

# 2) Observe the features

# 3) Drop 'ISO','Headcount Ratio Urban','Intensity of Deprivation Urban','Headcount Ratio Rural','Intensity of Deprivation Rural' columns and save the new dataset as 'my_mpi_ds'

#Rename the column heading as below
my_mpi_ds.rename(
    columns = {'Country':'country',
               'MPI Urban':'mpi_urban',
               'MPI Rural':'mpi_rural'
              },
    inplace = True)

# show the headings
my_mpi_ds.head(3)
```

Combine your original dataset (*Pov_data*) with *my_mpi_ds* and check the heading.

```
combined = pd.merge(
    Pov_data,
    my_mpi_ds,
    on='country',
    how='inner'
)

combined.head()
```

To Do 2: Perform correlation analysis on your new dataset ('combined'). Provide the correlation matrix output and explain your findings. Is there any multicollinearity within the features?

2.1. **Simple Linear Regression Model**

Create a simple linear regression model for all predictors and use 'mpi_urban' as the dependent variable.

```
reg = linear_model.LinearRegression()#linear regression class object

import statsmodels.api as sm
from statsmodels.formula.api import ols # libraries for plotting of
residual plots
```

Below code is example for predicting mpi_urban using child_mort as predictor.

```
#fit simple linear regression model
model = ols('mpi_urban ~ child_mort', data=combined).fit()

#print model summary
print(model.summary())

#adjust figure size
fig = plt.figure(figsize=(12,8))

#generate regression plots
fig = sm.graphics.plot_regress_exog(model, 'child_mort', fig=fig)
```

To Do 3: Create the model for remaining predictors and provide the results only. Your result report must be well formatted and readable.

Appendix A (MPI_Dataset Metadata)

Feature Description

- ISO: Unique ID for country
- Country: country name
- MPI Urban: Multi-dimensional poverty index for urban areas within the country
- Headcount Ratio Urban: Poverty headcount ratio (% of population listed as poor) within urban areas within the country
- Intensity of Deprivation Urban: Average distance below the poverty line of those listed as poor in urban areas
- MPI Rural: Multi-dimensional poverty index for rural areas within the country
- Headcount Ratio Rural: Poverty headcount ratio (% of population listed as poor) within rural areas within the country
- Intensity of Deprivation Rural: Average distance below the poverty line of those listed as poor in rural areas

For the purpose of this analysis we will focus on MPI Urban and MPI Rural. This is because the MPI measure reflects both:
a) the incidence of poverty (the percentage of the population who are poor) and,
b) the intensity of poverty (the percentage of deprivations suffered by each person or household on average). M0 is calculated by multiplying the incidence (H) by the intensity (A). M0 = H x A.