Predictive_Analysis_of_Productive_Employment_based_on_Economic

February 10, 2021

[ ]:
```
# Author: Isaac Mwendwa
# Reg. No: SCT211-0087/2016
# Computer Systems Project
# BSc. Computer Science 4.2
# SCIT -- JKUAT
```

[ ]:
```
import tensorflow as tf
print(tf.__version__)
```

```
2.4.1
```

[ ]:
```
!python -c "import sys; print(sys.version)"
```

```
3.6.9 (default, Oct  8 2020, 12:12:24)
[GCC 8.4.0]
```

0.1 Workflow of Project

This project utilizes the Data Science Project Life Cycle, which has the following steps:

1. Business Understanding
2. Data Collection
3. Data Preparation
4. Exploratory Data Analysis
5. Modelling
6. Model Evaluation
7. Model Deployment

1 Part 1: Business Understanding

1.0.1 Introduction to concepts of Economic Growth and Productive Employment

- Gross Domestic Product(GDP) as a measure of Economic Growth > * GDP = value of goods and services produced by the nations economy - value of goods and services used up in production > * Two measures of GDP will be used in this study, the Contribution_by_GDP and Growth_by_GDP

- Productive Employment > * Productive Employment is employment yielding sufficient returns to labour, to permit a worker and his/her dependents a level of consumption above the poverty line > * The International Labour Organization (ILO) has set the International Poverty Line to USD 2 (USD 1.90) a day; hence any person earning below USD 2 a day is considered poor, a group the ILO refers to as the 'Working Poor' > * The working poor in this case will be the people earning below KSh 10,000 a month (Wage_bracket_0_to_9999)

1.0.2 -- End of Business Understanding Section (Part 1)---

2 Part 2: Data Collection

2.0.1 Perform Data Extraction from PDF Files using camelot-py Module

2.0.2 Prerequisites

- Installing dependecies for camelot-py, which include GhostScript and TKinter, on local machine
- Installing camelot-py module
- Testing camelot-py module
- Downloading yearly Statistical Abstract files from Kenya National Bureau of Statitics (KNBS)

[ ]:
```

```

Sources of data (PDF files, courtesy of KNBS)

- Statistical Abstract 2013
- Statistical Abstract 2014
- Statistical Abstract 2015
- Statistical Abstract 2017
- Statistical Abstract 2019

To extract data from the pdf files, I connect to local runtime, which has camelot-py module installed; to use the module for extraction

2.0.3 Workflow

1. Set up notebook server to allow connection to local runtime; using the command provided in the next section
2. Look up for desired tables in pdf files, noting page numbers
3. Extract tables from the pdf files using the page numbers
4. Export tables to csv files, which will then be used in the data preparation section

[ ]:
```

```

[ ]:
```

```

Command for starting notebook server jupyter notebook --NotebookApp.allow_origin='https://colab.research.google.com' --port=8888 --NotebookApp.port_retries=0

Import camelot-py module

[ ]:
```
import camelot
```

Extracting tables from Statistical Abstract 2013

[ ]:
```
# Statistical Abstract 2013
pathPdf = 'F:/1Workspace/1Data/STATISTICAL ABSTRACT 2013.pdf'
pathDataset = 'F:/1Workspace/1Data/'
tables = camelot.read_pdf(pathPdf, pages='285', flavor='stream',␣
 ↪strip_text='*+')

numOfTables=tables.n
print('Number of tables: ' + str(numOfTables))

#Parsing Report
print('PARSING REPORT')
print(tables[0].parsing_report)
```

```
Number of tables: 1
PARSING REPORT
{'accuracy': 95.47, 'whitespace': 11.79, 'order': 1, 'page': 285}
```

[ ]:
```
tables[0].to_csv(pathDataset + '/Wage Employment 2011.csv')
```

Extracting tables from Statistical Abstract 2014

[ ]:
```
# Statistical Abstract 2014
pathPdf = 'F:/1Workspace/1Data/STATISTICAL-ABSTRACT-2014.pdf'
pathDataset = 'F:/1Workspace/1Data/'
tables = camelot.read_pdf(pathPdf, pages='74,77,265,266,267,268',␣
 ↪flavor='stream', strip_text='*+\n')

numOfTables=tables.n
print('Number of tables: ' + str(numOfTables))

#Parsing Report
print('PARSING REPORT')
i=0
while i < numOfTables:
    print(tables[i].parsing_report)
    i+=1
```

```
Number of tables: 6
PARSING REPORT
{'accuracy': 99.36, 'whitespace': 8.12, 'order': 1, 'page': 74}
{'accuracy': 99.63, 'whitespace': 10.68, 'order': 1, 'page': 77}
{'accuracy': 91.57, 'whitespace': 31.06, 'order': 1, 'page': 265}
{'accuracy': 97.76, 'whitespace': 14.48, 'order': 1, 'page': 266}
{'accuracy': 99.17, 'whitespace': 14.48, 'order': 1, 'page': 267}
{'accuracy': 99.65, 'whitespace': 18.93, 'order': 1, 'page': 268}
```

[ ]:
```
#Exporting Tables to CSV Files
tables[0].to_csv(pathDataset + '/Contribution to GDP by Percent 2009-2013.csv')
tables[1].to_csv(pathDataset + '/Growth  of GDP by Activity 2009-2013.csv')
tables[2].to_csv(pathDataset + '/Wage Employment 2010-2013.csv')
tables[3].to_csv(pathDataset + '/Wage Employment 2012.csv')
tables[4].to_csv(pathDataset + '/Wage Employment 2013.csv')
tables[5].to_csv(pathDataset + '/Wage Employment by Sex and Income 2010-2013.
 ↪csv')
```

Extracting tables from Statistical Abstract 2015

[ ]:
```
# Statistical Abstract 2015
pathPdf = 'F:/1Workspace/1Data/STATISTICAL ABSTRACT 2015.pdf'
pathDataset = 'F:/1Workspace/1Data/'
tables = camelot.read_pdf(pathPdf, pages='262', flavor='stream',␣
 ↪strip_text='*+')

numOfTables=tables.n
print('Number of tables: ' + str(numOfTables))

#Parsing Report
print('PARSING REPORT')
print(tables[0].parsing_report)
```

```
Number of tables: 1
PARSING REPORT
{'accuracy': 93.42, 'whitespace': 27.43, 'order': 1, 'page': 262}
```

[ ]:
```
tables[0].to_csv(pathDataset + '/Wage Employment 2014.csv')
```

Extracting tables from Statistical Abstract 2017

[ ]:
```
# Statistical Abstract 2017
pathPdf = 'F:/1Workspace/1Data/STATISTICAL ABSTRACT 2017.pdf'
pathDataset = 'F:/1Workspace/1Data/'
tables = camelot.read_pdf(pathPdf, pages='101,102', flavor='stream',␣
 ↪strip_text='*+\n')

numOfTables=tables.n
print('Number of tables: ' + str(numOfTables))
```

```
#Parsing Report
print('PARSING REPORT')
i=0
while i < numOfTables:
    print(tables[i].parsing_report)
    i+=1
```

```
Number of tables: 4
PARSING REPORT
{'accuracy': 93.23, 'whitespace': 19.68, 'order': 1, 'page': 101}
{'accuracy': 99.8, 'whitespace': 19.67, 'order': 2, 'page': 101}
{'accuracy': 93.23, 'whitespace': 19.68, 'order': 1, 'page': 102}
{'accuracy': 99.8, 'whitespace': 19.67, 'order': 2, 'page': 102}
```

[ ]:
```
tables[1].to_csv(pathDataset + '/Wage Employment 2015.csv')
tables[3].to_csv(pathDataset + '/Wage Employment 2016.csv')
```

Extracting tables from Statistical Abstract 2019

[ ]:
```
# Statistical Abstract 2019
pathPdf = 'F:/1Workspace/1Data/Statistical_Abstract_2019.pdf'
pathDataset = 'F:/1Workspace/1Data/'
tables = camelot.read_pdf(pathPdf, pages='30,32,63,64,65,66', flavor='stream',␣
 ↪strip_text='*+\n')

numOfTables=tables.n
print('Number of tables: ' + str(numOfTables))

#Parsing Report
print('PARSING REPORT')
i=0
while i < numOfTables:
    print(tables[i].parsing_report)
    i+=1
```

```
Number of tables: 6
PARSING REPORT
{'accuracy': 99.55, 'whitespace': 9.21, 'order': 1, 'page': 30}
{'accuracy': 99.62, 'whitespace': 8.97, 'order': 1, 'page': 32}
{'accuracy': 88.77, 'whitespace': 29.29, 'order': 1, 'page': 63}
{'accuracy': 99.95, 'whitespace': 17.24, 'order': 1, 'page': 64}
{'accuracy': 99.96, 'whitespace': 19.67, 'order': 1, 'page': 65}
{'accuracy': 99.44, 'whitespace': 19.71, 'order': 1, 'page': 66}
```

[ ]:
```
#Exporting Tables to CSV Files
tables[0].to_csv(pathDataset + '/Contribution to GDP by Percent 2012-2018.csv')
tables[1].to_csv(pathDataset + '/Growth  of GDP by Activity 2012-2018.csv')
tables[2].to_csv(pathDataset + '/Wage Employment 2014-2018.csv')
```

```
tables[3].to_csv(pathDataset + '/Wage Employment 2017.csv')
tables[4].to_csv(pathDataset + '/Wage Employment 2018.csv')
tables[5].to_csv(pathDataset + '/Wage Employment by Sex and Income 2014-2018.
 ↪csv')
```

After performing data collection, we can now disconnect local runtime and switch to hosted runtime for the next sections

2.0.4 --- End of Data Collection Section (Part 2)---

[ ]:
```

```

3 Part 3: Data Preparation

3.0.1 Prerequisites

1. Preparing datasets using Microsoft Excel
2. Connection to hosted runtime
3. Migrating prepared datasets from local disk to Google Drive
4. Mounting Google Drive

3.0.2 Workflow

1. Joining yearly datasets to a single dataset spanning all the years (2011 -2018)
2. Data Pre-processing

3.0.3 Mounting Google Drive

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

3.0.4 Importing all packages needed

[ ]:
```
#imports
import sys
import numpy as np # linear algebra
from scipy.stats import randint
import pandas as pd # data processing, CSV file I/O
import matplotlib.pyplot as plt # this is used for the plot the graph
import seaborn as sns # used for plot interactive graph.
```

3.1 Joining yearly datasets to a single dataset spanning all the years

[ ]:
```
df2011 = pd.read_csv('/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2011.csv')
df2012 = pd.read_csv('/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2012.csv')
df2013 = pd.read_csv('/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2013.csv')
df2014 = pd.read_csv('/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2014.csv')
df2015 = pd.read_csv('/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2015.csv')
df2016 = pd.read_csv('/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2016.csv')
df2017 = pd.read_csv('/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2017.csv')
df2018 = pd.read_csv('/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2018.csv')
```

[ ]:
```
#Joining datasets along the row
pathDataset = '/content/drive/My Drive/ColabNotebooks/Project/'
df = df2011.append([df2012, df2013, df2014, df2015, df2016, df2017,␣
↪df2018],ignore_index=True, sort=False)
df.to_csv(pathDataset + 'Wage_Employment_and_GDP_2011_to_2018.csv')
```

3.0.5 3.2 Data Preprocessing

[ ]:
```
import pandas as pd

#path to dataset
pathDataset = '/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2011_to_2018.csv'
```

[ ]:
```
#Reading dataset
df = pd.read_csv(pathDataset)
df.head()
```

[ ]:
```
   Unnamed: 0  ...    TOTAL
0           0  ...  341,422
1           1  ...    8,732
2           2  ...  276,885
3           3  ...   12,338
4           4  ...    7,890

[5 rows x 14 columns]
```

[ ]:
```
df.columns
```

[ ]:
```
Index(['Unnamed: 0', 'Industry', 'Year', 'Contribution_by_Gdp',
       'Growth_of_GDP', '0 - 9,999', '10,000 - 14999', '15,000 - 19999',
```

```
       '20,000 - 24999', '25,000 - 29999', '30,000 - 49999', '50,000 - 99999',
       '100000+', 'TOTAL'],
      dtype='object')
```

[ ]:
```
df.drop('Unnamed: 0', axis=1, inplace=True)
```

Renaming columns

[ ]:
```
df.columns = ['Industry', 'Year', 'Contribution_to_GDP', 'Growth_of_GDP',
       'Wage_bracket_0_to_9999', 'Wage_bracket_10000_to_14999',
       'Wage_bracket_15000_to_19999', 'Wage_bracket_20000_to_24999',
       'Wage_bracket_25000_to_29999', 'Wage_bracket_30000_to_49999',
       'Wage_bracket_50000_to_99999', 'Wage_bracket_100000_plus', 'TOTAL']
```

[ ]:
```
df.head()
```

[ ]:
```
                                    Industry  ...    TOTAL
0              Agriculture, Forestry And Fishing  ...  341,422
1                           Mining And Quarrying  ...    8,732
2                                  Manufacturing  ...  276,885
3  Electricity, Gas, Steam And Air Conditioning S...  ...   12,338
4  Water Supply; Sewerage, Waste Management And R...  ...    7,890

[5 rows x 13 columns]
```

Removing special characters from Wage_bracket columns

[ ]:
```
cols = ['Wage_bracket_0_to_9999', 'Wage_bracket_10000_to_14999',
       'Wage_bracket_15000_to_19999', 'Wage_bracket_20000_to_24999',
       'Wage_bracket_25000_to_29999', 'Wage_bracket_30000_to_49999',
       'Wage_bracket_50000_to_99999', 'Wage_bracket_100000_plus', 'TOTAL']

df[cols] = df[cols].astype(str)  # cast to string

# Removing special characters
df[cols] = df[cols].replace({'\$': '', ',': '', '-': ''}, regex=True)

# Renaming Contribution_by_GDP column
#df.rename(columns = {'Contribution_by_Gdp':'Contribution_to_GDP'}, inplace =␣
 ↪True)

# path to dataset
path = '/content/drive/My Drive/ColabNotebooks/Project/'
df.to_csv(path + 'Wage_Employment_and_GDP_2011_to_2018_Final.csv')
```

Reading sanitized dataset

[ ]:
```
pathDataset = '/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2011_to_2018_Final.csv'
df = pd.read_csv(pathDataset,
                 parse_dates=['Year'],
                 index_col=['Year'],
                 na_values=['nan','?','-'])
#df = df.set_index(['Year']) # Setting index to Year
#df.index = pd.to_datetime(df.index, format='%Y')  # Converting index to␣
↪datetime
df
```

[ ]:
```
            Unnamed: 0  ...     TOTAL
Year                    ...
2011-01-01           0  ...  341422.0
2011-01-01           1  ...    8732.0
2011-01-01           2  ...  276885.0
2011-01-01           3  ...   12338.0
2011-01-01           4  ...    7890.0
...                ...  ...       ...
2018-01-01         163  ...  148755.0
2018-01-01         164  ...    7243.0
2018-01-01         165  ...   36332.0
2018-01-01         166  ...       NaN
2018-01-01         167  ...  115836.0

[168 rows x 13 columns]
```

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 168 entries, 2011-01-01 to 2018-01-01
Data columns (total 13 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Unnamed: 0                  168 non-null    int64
 1   Industry                    168 non-null    object
 2   Contribution_to_GDP         166 non-null    float64
 3   Growth_of_GDP               161 non-null    float64
 4   Wage_bracket_0_to_9999      96 non-null     float64
 5   Wage_bracket_10000_to_14999 144 non-null    float64
 6   Wage_bracket_15000_to_19999 152 non-null    float64
 7   Wage_bracket_20000_to_24999 167 non-null    float64
 8   Wage_bracket_25000_to_29999 167 non-null    float64
 9   Wage_bracket_30000_to_49999 167 non-null    float64
 10  Wage_bracket_50000_to_99999 167 non-null    float64
 11  Wage_bracket_100000_plus    154 non-null    float64
 12  TOTAL                       167 non-null    float64
dtypes: float64(11), int64(1), object(1)
```

```
memory usage: 18.4+ KB
```

[ ]:
```
df.dtypes
```

[ ]:
```
Unnamed: 0                     int64
Industry                      object
Contribution_to_GDP          float64
Growth_of_GDP                float64
Wage_bracket_0_to_9999       float64
Wage_bracket_10000_to_14999  float64
Wage_bracket_15000_to_19999  float64
Wage_bracket_20000_to_24999  float64
Wage_bracket_25000_to_29999  float64
Wage_bracket_30000_to_49999  float64
Wage_bracket_50000_to_99999  float64
Wage_bracket_100000_plus     float64
TOTAL                        float64
dtype: object
```

[ ]:
```
df.shape
```

[ ]:
```
(168, 13)
```

[ ]:
```
df.columns
```

[ ]:
```
Index(['Unnamed: 0', 'Industry', 'Contribution_to_GDP', 'Growth_of_GDP',
       'Wage_bracket_0_to_9999', 'Wage_bracket_10000_to_14999',
       'Wage_bracket_15000_to_19999', 'Wage_bracket_20000_to_24999',
       'Wage_bracket_25000_to_29999', 'Wage_bracket_30000_to_49999',
       'Wage_bracket_50000_to_99999', 'Wage_bracket_100000_plus', 'TOTAL'],
      dtype='object')
```

3.0.6 Dealing with nan values -- filling nan with mean in the columns

[ ]:
```
# finding all columns that have nan:
nan_list =[]  #list of columns with nan values
for j in range(2,13):
    if not df.iloc[:, j].notnull().all():
        nan_list.append(j)
nan_list
```

[ ]:
```
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

[ ]:
```
# filling nan with mean in any columns
for j in range(2,13):
        df.iloc[:,j]=df.iloc[:,j].fillna(df.iloc[:,j].mean())
```

Verifying no column has NaN now

[ ]:
```
# Checking if any column has nan
df.isnull().sum()
```

[ ]:
```
Unnamed: 0                    0
Industry                      0
Contribution_to_GDP           0
Growth_of_GDP                 0
Wage_bracket_0_to_9999        0
Wage_bracket_10000_to_14999   0
Wage_bracket_15000_to_19999   0
Wage_bracket_20000_to_24999   0
Wage_bracket_25000_to_29999   0
Wage_bracket_30000_to_49999   0
Wage_bracket_50000_to_99999   0
Wage_bracket_100000_plus      0
TOTAL                         0
dtype: int64
```

Checking dtypes of columns

[ ]:
```
df.dtypes
```

[ ]:
```
Unnamed: 0                       int64
Industry                        object
Contribution_to_GDP            float64
Growth_of_GDP                  float64
Wage_bracket_0_to_9999         float64
Wage_bracket_10000_to_14999    float64
Wage_bracket_15000_to_19999    float64
Wage_bracket_20000_to_24999    float64
Wage_bracket_25000_to_29999    float64
Wage_bracket_30000_to_49999    float64
Wage_bracket_50000_to_99999    float64
Wage_bracket_100000_plus       float64
TOTAL                          float64
dtype: object
```

Casting Wage_bracket_ columns from float64 to int64

[ ]:
```
cols = ['Wage_bracket_0_to_9999', 'Wage_bracket_10000_to_14999',
       'Wage_bracket_15000_to_19999', 'Wage_bracket_20000_to_24999',
       'Wage_bracket_25000_to_29999', 'Wage_bracket_30000_to_49999',
       'Wage_bracket_50000_to_99999', 'Wage_bracket_100000_plus']


df[cols] = df[cols].astype(int)  # cast to int
```

Confirming that the dtypes of Wage_bracket_ columns have been converted from float to int

[ ]:
```
df.dtypes
#df.head()
```

[ ]:
```
Unnamed: 0                      int64
Industry                       object
Contribution_to_GDP           float64
Growth_of_GDP                 float64
Wage_bracket_0_to_9999          int64
Wage_bracket_10000_to_14999     int64
Wage_bracket_15000_to_19999     int64
Wage_bracket_20000_to_24999     int64
Wage_bracket_25000_to_29999     int64
Wage_bracket_30000_to_49999     int64
Wage_bracket_50000_to_99999     int64
Wage_bracket_100000_plus        int64
TOTAL                         float64
dtype: object
```

[ ]:
```
df.describe()
```

[ ]:
```
       Unnamed: 0  Contribution_to_GDP  ...  Wage_bracket_100000_plus
TOTAL
count  168.000000           168.000000  ...                168.000000
168.000000
mean    83.500000             4.403012  ...               4186.000000
116453.365269
std     48.641546             6.463415  ...               6741.704946
127518.568458
min      0.000000            -2.800000  ...                 13.000000
1009.000000
25%     41.750000             0.800000  ...                540.000000
13902.250000
50%     83.500000             1.700000  ...               2585.000000
76407.500000
75%    125.250000             6.725000  ...               5052.750000
164204.250000
max    167.000000            34.800000  ...              56221.000000
576831.000000

[8 rows x 12 columns]
```

[ ]:
```
df.columns
```

[ ]:
```
Index(['Unnamed: 0', 'Industry', 'Contribution_to_GDP', 'Growth_of_GDP',
       'Wage_bracket_0_to_9999', 'Wage_bracket_10000_to_14999',
       'Wage_bracket_15000_to_19999', 'Wage_bracket_20000_to_24999',
       'Wage_bracket_25000_to_29999', 'Wage_bracket_30000_to_49999',
       'Wage_bracket_50000_to_99999', 'Wage_bracket_100000_plus', 'TOTAL'],
      dtype='object')
```

[ ]:

Adding new column for total_number_in_wage_employment To accomodate changes made in wage_bracket columns, we need to have a column for the new total number of people in wage_bracket columns

[ ]:
```
# Removing unwanted columns, to remain with wage_bracket_cols
col_list= list(df)
unwanted = {'Unnamed: 0', 'Industry', 'Contribution_to_GDP', 'Growth_of_GDP',␣
↪'TOTAL'} # columns to remove
wage_bracket_cols = [e for e in col_list if e not in unwanted] # Removing␣
↪columns in unwanted
wage_bracket_cols
```

[ ]:
```
['Wage_bracket_0_to_9999',
 'Wage_bracket_10000_to_14999',
 'Wage_bracket_15000_to_19999',
 'Wage_bracket_20000_to_24999',
 'Wage_bracket_25000_to_29999',
 'Wage_bracket_30000_to_49999',
 'Wage_bracket_50000_to_99999',
 'Wage_bracket_100000_plus']
```

[ ]:
```
#  Adding new column for total, which is a sum of rows of wage_bracket_cols
df['Total_number_in_wage_employment'] = df[wage_bracket_cols].sum(axis=1)
df.head()
```

[ ]:
```
            Unnamed: 0  ... Total_number_in_wage_employment
Year                    ...
2011-01-01           0  ...                          341422
2011-01-01           1  ...                            8732
2011-01-01           2  ...                          276885
2011-01-01           3  ...                           14018
2011-01-01           4  ...                           21211

[5 rows x 14 columns]
```

[ ]:

3.0.7 Deleting old total column, saving dataframe to csv (updated dataset)

[ ]:
```
#Deleting old total column
df.drop('TOTAL', axis=1, inplace=True)

#Saving dataset
path = '/content/drive/My Drive/ColabNotebooks/Project/'  #path to dataset
df.to_csv(path + 'Wage_Employment_and_GDP_2011_to_2018_Updated.csv')  #saving␣
↪updated dataset
```

3.0.8 -- End of Data Preparation Section (Part 3) ------------------------

---

[ ]:

```

```

4 Part 4: Exploratory Data Analysis

4.0.1 Workflow

1. Reading updated dataset
2. Performing data exploration

4.0.2 Reading updated dataset

[ ]:

```
pathDataset = '/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2011_to_2018_Updated.csv'
df = pd.read_csv(pathDataset, parse_dates=['Year'],
                 index_col=['Year'],)
df.head()
```

[ ]:

```
            Unnamed: 0  ... Total_number_in_wage_employment
Year                    ...
2011-01-01           0  ...                          341422
2011-01-01           1  ...                            8732
2011-01-01           2  ...                          276885
2011-01-01           3  ...                           14018
2011-01-01           4  ...                           21211

[5 rows x 13 columns]
```

[ ]:

```
# Confirming updated dataset doesn't have nan
df.isnull().sum()
```

[ ]:

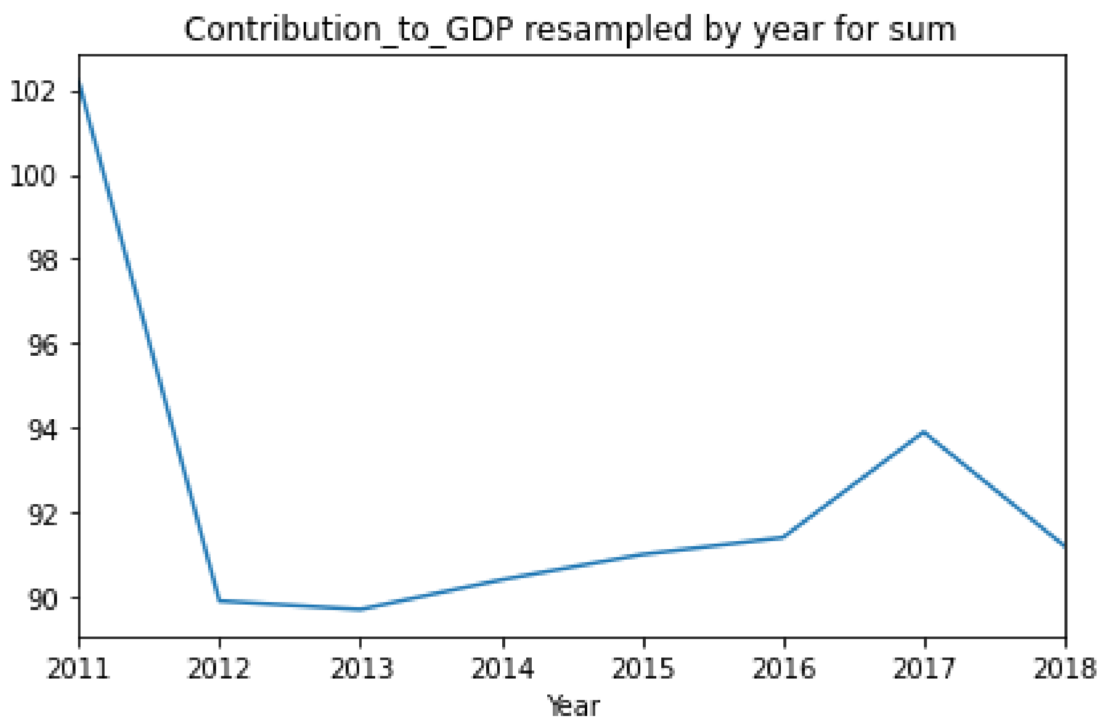
```
Unnamed: 0                         0
Industry                           0
Contribution_to_GDP                0
Growth_of_GDP                      0
Wage_bracket_0_to_9999             0
Wage_bracket_10000_to_14999        0
Wage_bracket_15000_to_19999        0
Wage_bracket_20000_to_24999        0
Wage_bracket_25000_to_29999        0
Wage_bracket_30000_to_49999        0
Wage_bracket_50000_to_99999        0
Wage_bracket_100000_plus           0
```

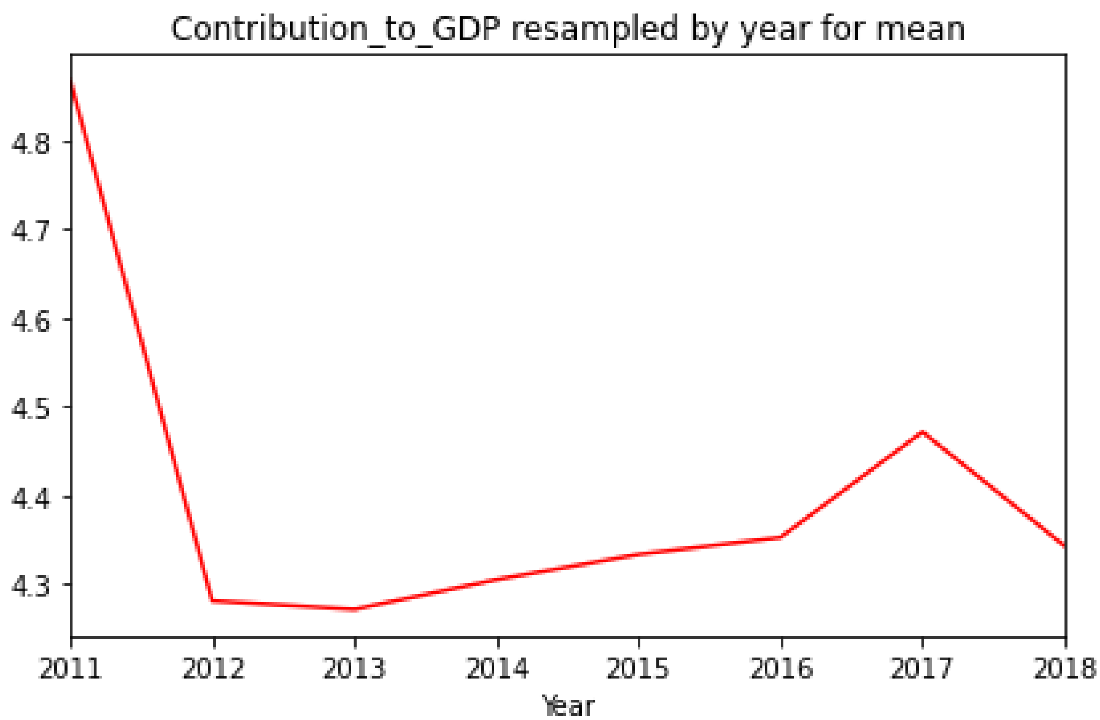
```
Total_number_in_wage_employment    0
dtype: int64
```

[ ]:
```
#df.describe()
```

4.0.3 Resampling Contribution_to_GDP for sum and mean by year

[ ]:
```
# Resampling the sum of Contribution_to_GDP by Year
df.Contribution_to_GDP.resample('Y').sum().plot(title = 'Contribution_to_GDP␣
 ↪resampled by year for sum')
plt.tight_layout()
plt.show()

# Resampling the mean of Contribution_to_GDP by Year
df.Contribution_to_GDP.resample('Y').mean().plot(title = 'Contribution_to_GDP␣
 ↪resampled by year for mean', color='red')
plt.tight_layout()
plt.show()
```

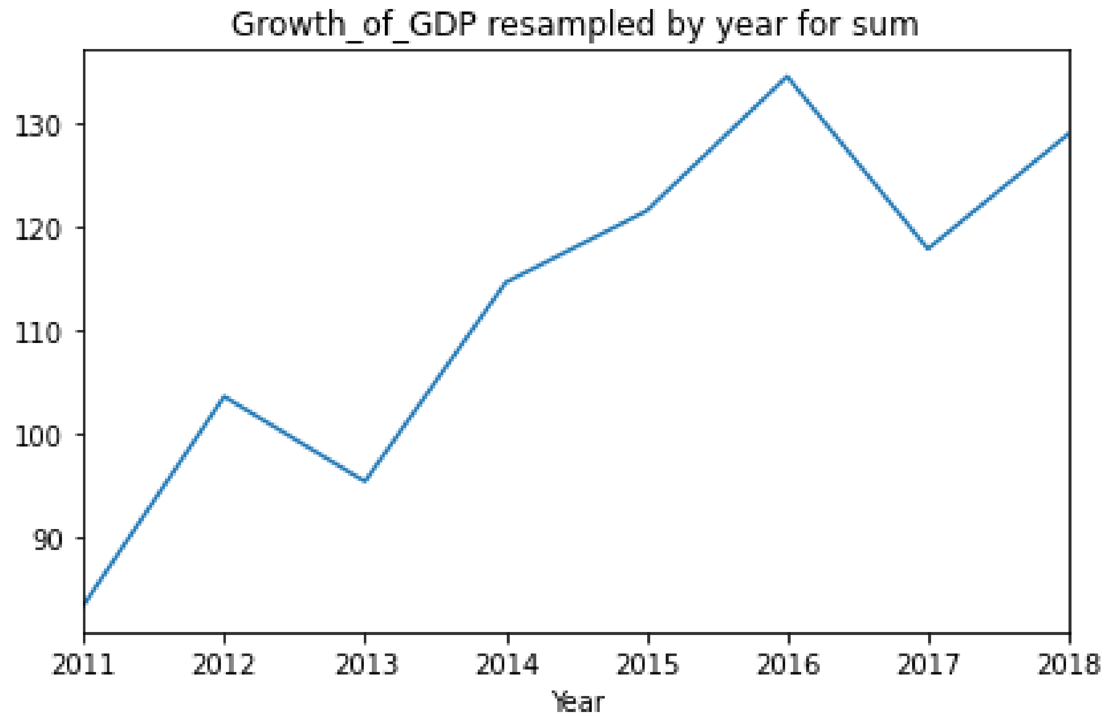


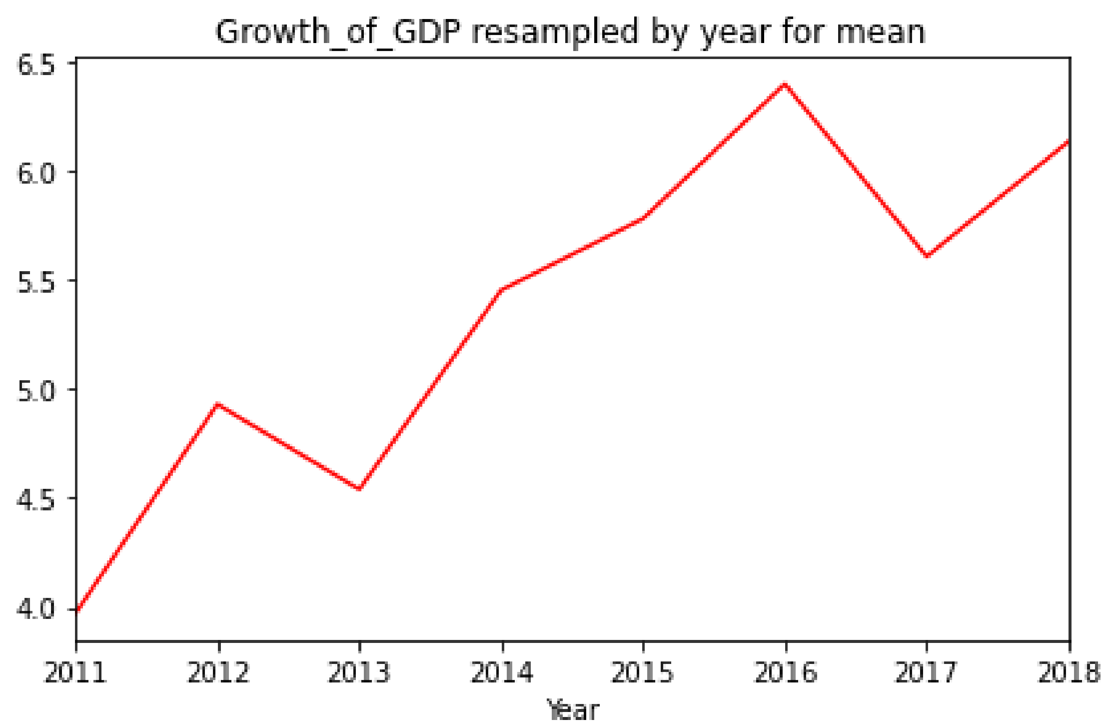


- The above plots show that resampling Contribution_to_GDP by year, for both sum and mean; yields similar plots, hence similar structure of dataset

4.0.4 Resampling Growth_of_GDP for sum and mean by year

[ ]:
```
# Resampling the sum of Contribution_to_GDP by Year
df.Growth_of_GDP.resample('Y').sum().plot(title = 'Growth_of_GDP resampled by␣
 ↪year for sum')
plt.tight_layout()
plt.show()

# Resampling the mean of Contribution_to_GDP by Year
df.Growth_of_GDP.resample('Y').mean().plot(title = 'Growth_of_GDP resampled by␣
 ↪year for mean', color='r')
plt.tight_layout()
plt.show()
```


Growth_of_GDP resampled by year for sum
130
120
110
100
90
2011
2012
2013
2014
2015
2016
2017
2018
Year


Growth_of_GDP resampled by year for mean
6.5
6.0
5.5
5.0
4.5
4.0
2011
2012
2013
2014
2015
2016
2017
2018
Year

[ ]:
```
df.dtypes
```

[ ]:
```
Unnamed: 0                          int64
Industry                           object
Contribution_to_GDP               float64
Growth_of_GDP                     float64
Wage_bracket_0_to_9999              int64
Wage_bracket_10000_to_14999         int64
Wage_bracket_15000_to_19999         int64
Wage_bracket_20000_to_24999         int64
Wage_bracket_25000_to_29999         int64
Wage_bracket_30000_to_49999         int64
Wage_bracket_50000_to_99999         int64
Wage_bracket_100000_plus            int64
Total_number_in_wage_employment     int64
dtype: object
```

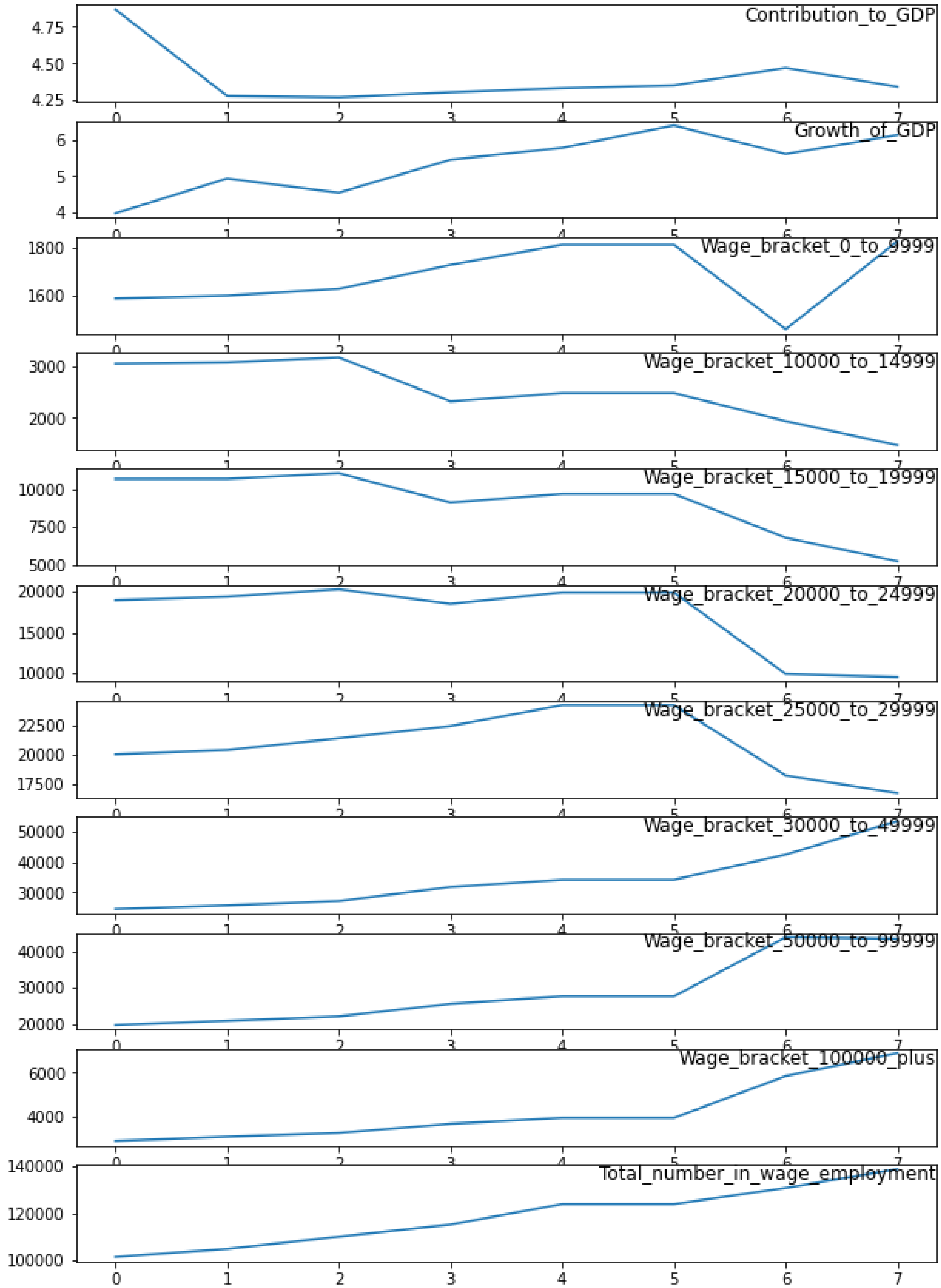
[ ]:
```
# Creating df1 -- dataframe for data columns
df1=df
df1.drop('Industry', axis=1, inplace=True)
df1.drop('Unnamed: 0', axis=1, inplace=True)
df1.head()
```

[ ]:
```
            Contribution_to_GDP  ...  Total_number_in_wage_employment
Year                             ...
2011-01-01                 23.8  ...                           341422
2011-01-01                  0.7  ...                             8732
2011-01-01                  9.6  ...                           276885
2011-01-01                  0.4  ...                            14018
2011-01-01                  0.7  ...                            21211

[5 rows x 11 columns]
```

4.0.5 Resampling all data columns for mean by year

[ ]:
```
# Specifying columns
cols = [0,1,2,3,4,5,6,7,8,9,10]
i=1
#groups=cols
df1=df

values = df1.resample('Y').mean().values
# plot each column
plt.figure(figsize=(10, 15))
for col in cols:
  plt.subplot(len(cols), 1, i)
  plt.plot(values[:, col])
  plt.title(df.columns[col], y=0.75, loc='right')
```

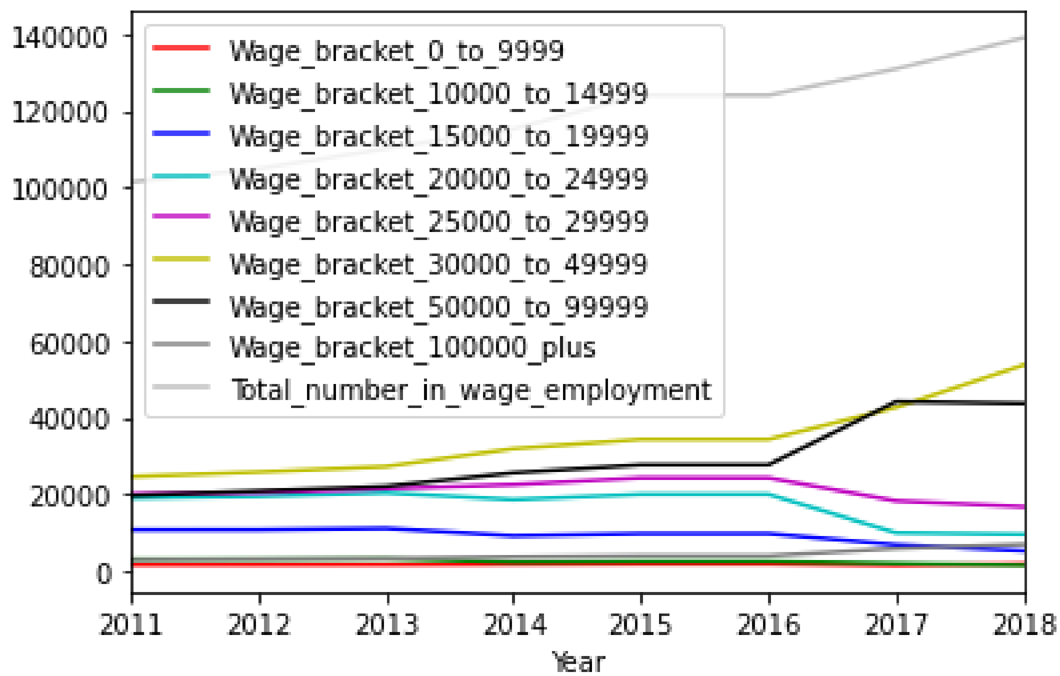
```
    i += 1
plt.show()
```



Contribution_to_GDP
4.75
4.50
4.25
Growth_of_GDP
6
5
4
Wage_bracket_0_to_9999
1800
1600
Wage_bracket_10000_to_14999
3000
2000
Wage_bracket_15000_to_19999
10000
7500
5000
Wage_bracket_20000_to_24999
20000
15000
10000
Wage_bracket_25000_to_29999
22500
20000
17500
Wage_bracket_30000_to_49999
50000
40000
30000
Wage_bracket_50000_to_99999
40000
30000
20000
Wage_bracket_100000_plus
6000
4000
Total_number_in_wage_employment
140000
120000
100000
0 1 2 3 4 5 6 7

[ ]:
```
#df.Contribution_to_GDP.resample('Y').mean().plot(color='r', legend=True)
#df.Growth_of_GDP.resample('Y').mean().plot(color='g', legend=True)
df.Wage_bracket_0_to_9999.resample('Y').mean().plot(color='r', legend=True)
df.Wage_bracket_10000_to_14999.resample('Y').mean().plot(color='g', legend=True)
df.Wage_bracket_15000_to_19999.resample('Y').mean().plot(color='b', legend=True)
df.Wage_bracket_20000_to_24999.resample('Y').mean().plot(color='c', legend=True)
df.Wage_bracket_25000_to_29999.resample('Y').mean().plot(color='m', legend=True)
df.Wage_bracket_30000_to_49999.resample('Y').mean().plot(color='y', legend=True)
df.Wage_bracket_50000_to_99999.resample('Y').mean().plot(color='k', legend=True)
df.Wage_bracket_100000_plus.resample('Y').mean().plot(color='0.5', legend=True)
df.Total_number_in_wage_employment.resample('Y').mean().plot(color='0.75',␣
↪legend=True)
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba20c8d630>
```



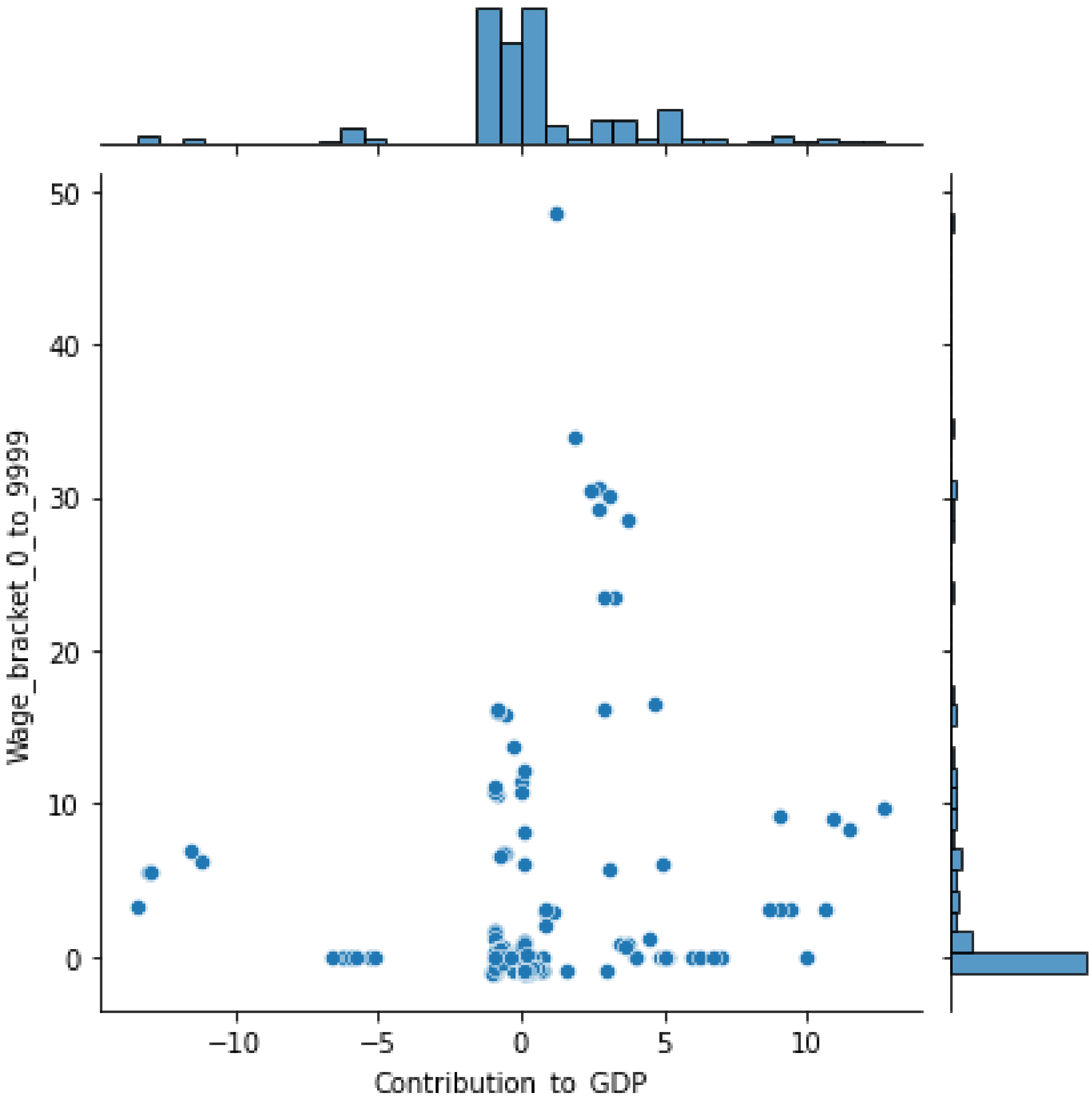
Correlation between 'Contribution_to_GDP', 'Wage_bracket_0_to_9999'

[ ]:
```
data = df [['Contribution_to_GDP','Wage_bracket_0_to_9999']]
correlation = data.corr(method='pearson')
correlation
```

[ ]:

	Contribution_to_GDP	Wage_bracket_0_to_9999
Contribution_to_GDP	1.000000	0.748173
Wage_bracket_0_to_9999	0.748173	1.000000

[ ]:
```
## The correlations between 'Contribution_to_GDP', 'Wage_bracket_0_to_9999'
data_returns = df.pct_change()
sns.jointplot(x='Contribution_to_GDP', y='Wage_bracket_0_to_9999',␣
↪data=data_returns)

plt.show()
```

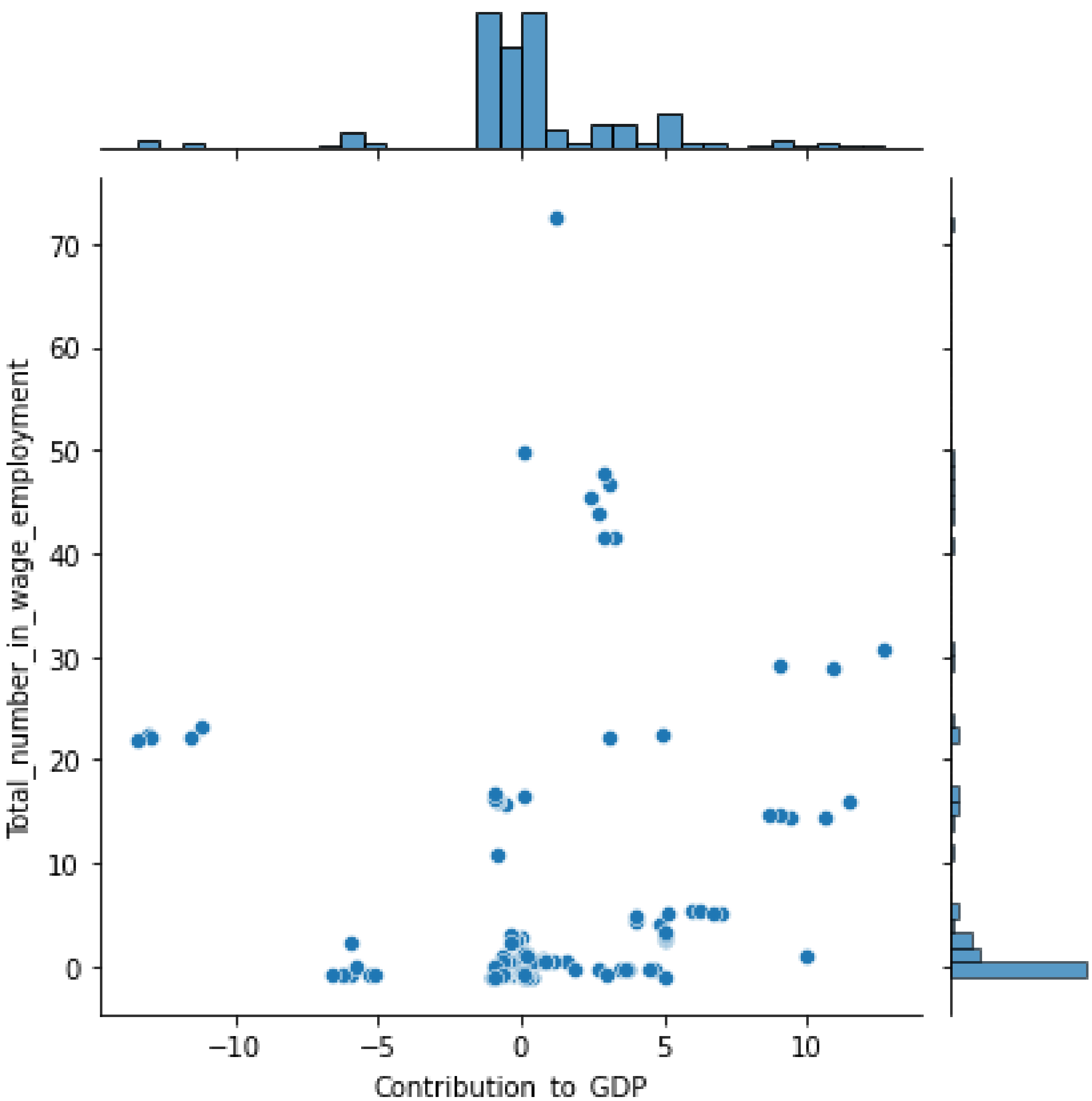


Correlation between 'Contribution_to_GDP', 'Total_number_in_wage_employment'

[ ]:
```
data = df[['Contribution_to_GDP', 'Total_number_in_wage_employment']]
correlation = data.corr(method='pearson')
correlation
```

[ ]:
```
                                      Contribution_to_GDP
Total_number_in_wage_employment
Contribution_to_GDP                               1.00000
0.52969
Total_number_in_wage_employment                   0.52969
1.00000
```

[ ]:
```
sns.jointplot(x='Contribution_to_GDP', y='Total_number_in_wage_employment',␣
 ↪data=data_returns)
plt.show()
```



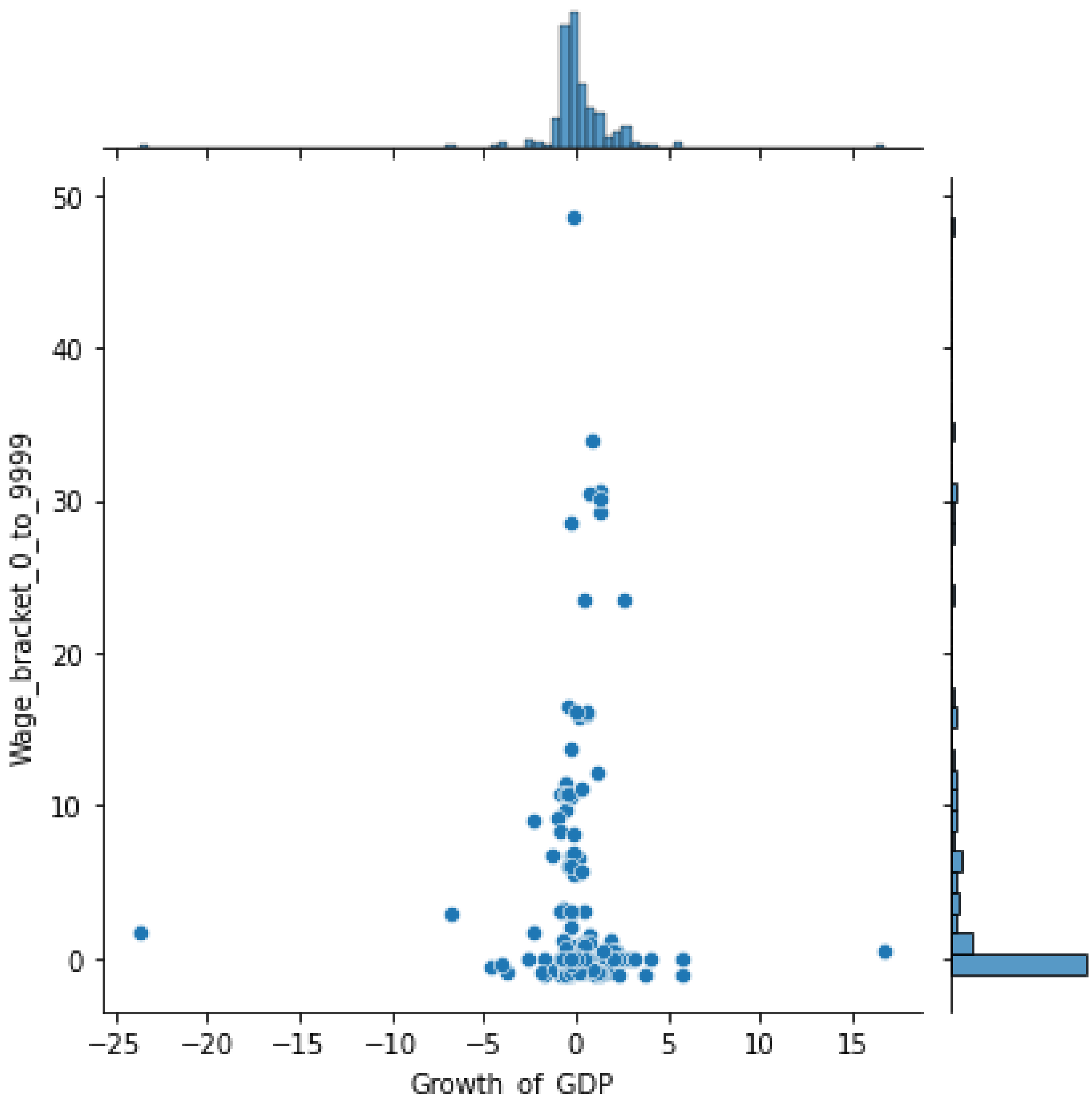
Correlation between 'Growth_of_GDP', 'Wage_bracket_0_to_9999'

[ ]:
```
data = df[['Growth_of_GDP', 'Wage_bracket_0_to_9999']]
correlation = data.corr(method='pearson')
correlation
```

[ ]:

	Growth_of_GDP	Wage_bracket_0_to_9999
Growth_of_GDP	1.000000	-0.080076
Wage_bracket_0_to_9999	-0.080076	1.000000

[ ]:
```
sns.jointplot(x='Growth_of_GDP', y='Wage_bracket_0_to_9999', data=data_returns)
plt.show()
```



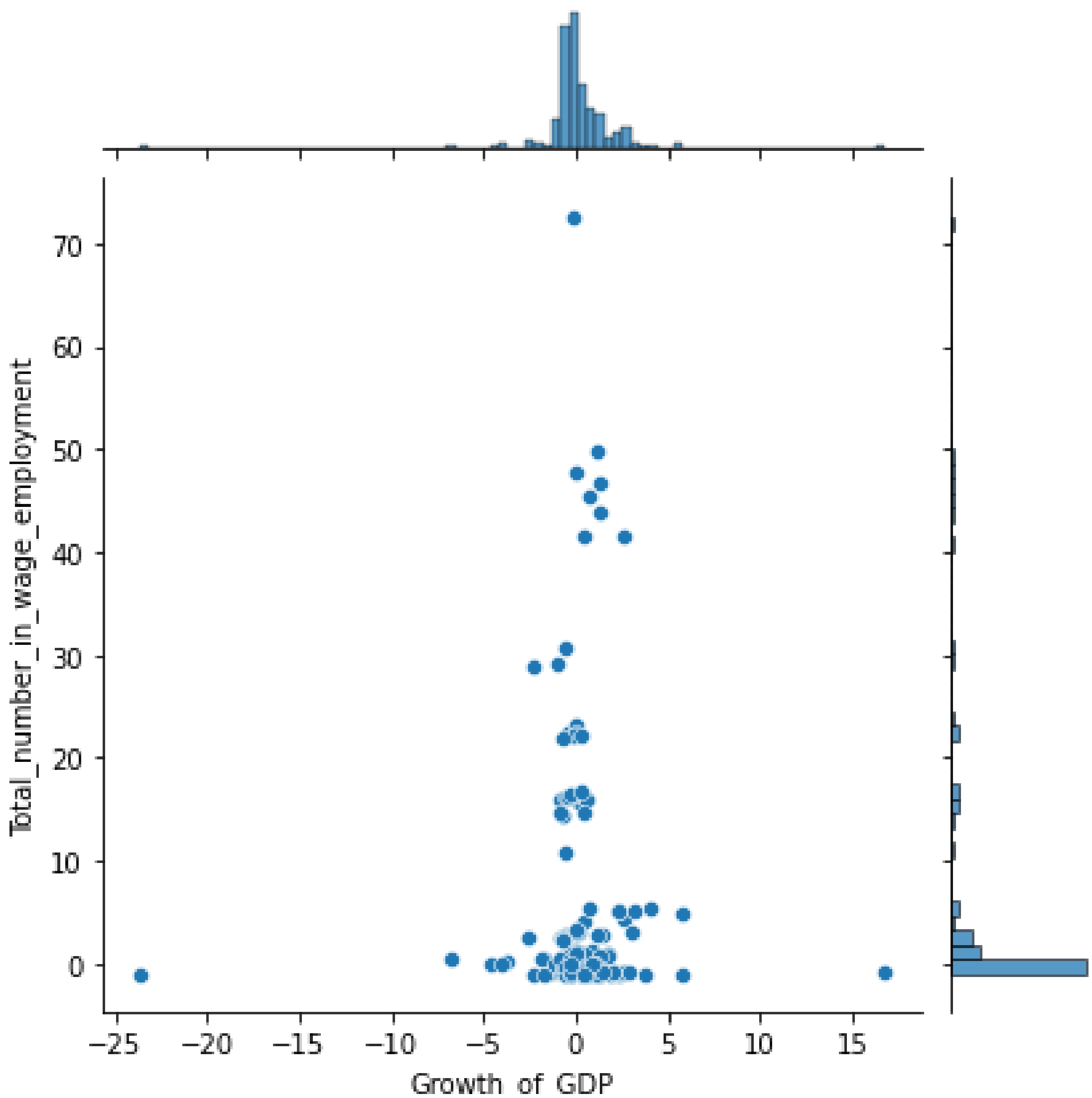
Correlation between 'Growth_of_GDP', 'Total_number_in_wage_employment'

[ ]:
```
data = df[['Growth_of_GDP', 'Total_number_in_wage_employment']]
correlation = data.corr(method='pearson')
correlation
```

[ ]:

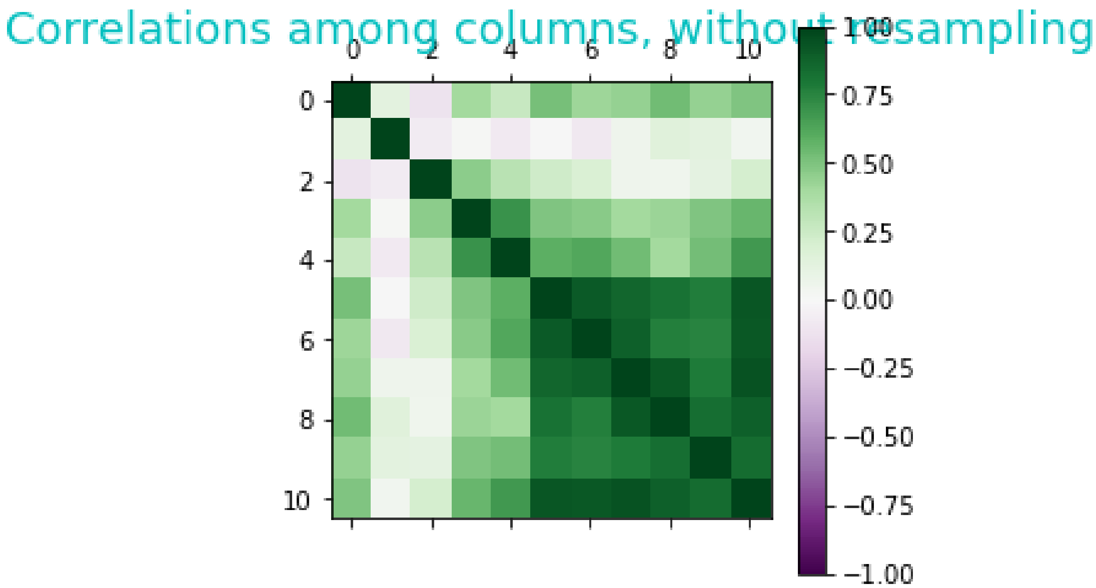
	Growth_of_GDP	Total_number_in_wage_employment
Growth_of_GDP	1.00000	-0.01246
Total_number_in_wage_employment	-0.01246	1.00000

[ ]:
```
sns.jointplot(x='Growth_of_GDP', y='Total_number_in_wage_employment',␣
 ↪data=data_returns)
plt.show()
```

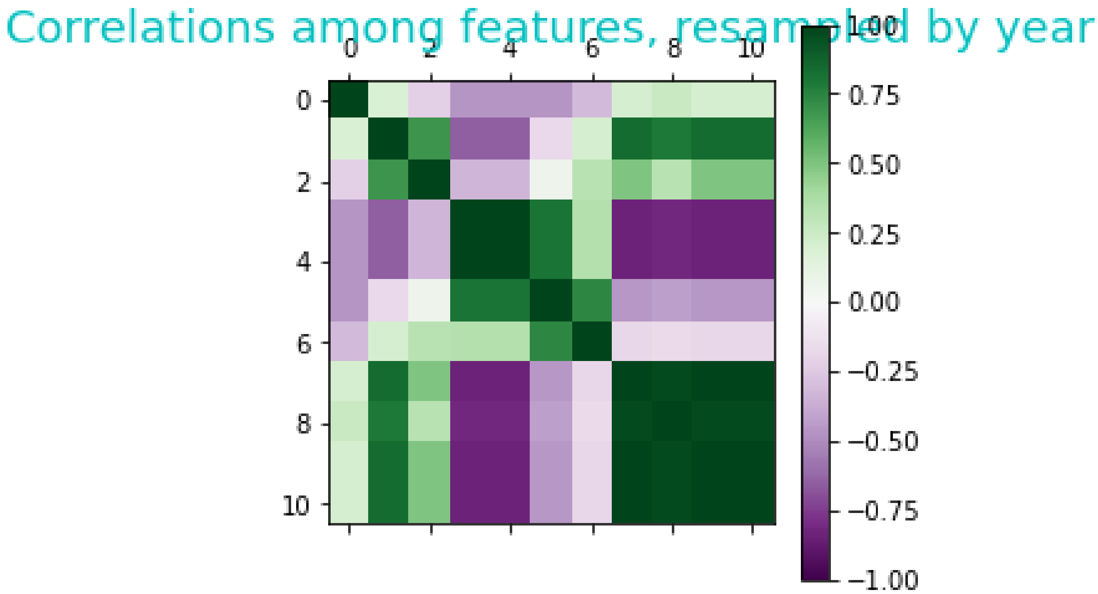


4.0.6 Correlations among features

[ ]:
```
# Correlations among columns, without resampling
plt.matshow(df.corr(method='spearman'),vmax=1,vmin=-1,cmap='PRGn')
plt.title('Correlations among columns, without resampling', size=18, color='c')
plt.colorbar()
plt.show()
```



[ ]:
```
# Correlations of mean of features resampled by year
plt.matshow(df.resample('Y').mean().
 ↪corr(method='spearman'),vmax=1,vmin=-1,cmap='PRGn')
plt.title('Correlations among features, resampled by year', size=18, color='c')
plt.colorbar()
plt.show()
```



- The above plots show that resampling techniques allow one to be able to make changes to the correlations among features.
- This is crucial to the feature engineering step in model building

4.0.7 --- End of Part 4(Exploratory Data Analysis) ----------- ------------------------

---

[ ]:

```

```

5 Part 5: Modelling

In this section, I develop two models -- one for the predictive analysis of the working poor (Wage_bracket_0_to_9999) and the other for predictive analysis of total employment (Total_number_in_wage_employment)

5.1 Workflow

1. LSTM Data Preparation
2. Defination of supervised learning problem
3. Modelling

5.1.1 1. LSTM Data Preparation

[ ]:
```
#importing packages needed
from math import sqrt
from numpy import concatenate
import matplotlib.pyplot as plt
from pandas import read_csv, get_dummies
from pandas import DataFrame
from pandas import concat
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM

# convert series to supervised learning
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
        n_vars = 1 if type(data) is list else data.shape[1]
        df = DataFrame(data)
        cols, names = list(), list()
        # input sequence (t-n, ... t-1)
        for i in range(n_in, 0, -1):
                cols.append(df.shift(i))
                names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
        # forecast sequence (t, t+1, ... t+n)
        for i in range(0, n_out):
                cols.append(df.shift(-i))
                if i == 0:
                        names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
                else:
                        names += [('var%d(t+%d)' % (j+1, i)) for j in␣
 ↪range(n_vars)]
        # put it all together
        agg = concat(cols, axis=1)
        agg.columns = names
        # drop rows with NaN values
        if dropnan:
                agg.dropna(inplace=True)
        return agg
```

5.1.2 Mounting Google Drive

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

5.2 ------------ Model 1: Predictive Analysis of the Working Poor (Wage_bracket_0_to_9999) ----------------

---

5.2.1 Defination of Problem

I define the supervised learning problem as predicting the number of people earning less than USD 2 a day(International Poverty Line), which lie in *Wage_bracket_0_to_9999;* at the current year (t), given the GDP indicators and other inputs at the prior time step

5.2.2 Loading Dataset

[ ]:
```
#import pandas as pd

# load dataset
pathDataset = '/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2011_to_2018_Updated.csv'
dataset = read_csv(pathDataset, header=0, index_col=0)
#dataset.drop('Industry', axis=1, inplace=True)
dataset.drop('Unnamed: 0', axis=1, inplace=True)
print(dataset.columns)
```

```
Index(['Industry', 'Contribution_to_GDP', 'Growth_of_GDP',
       'Wage_bracket_0_to_9999', 'Wage_bracket_10000_to_14999',
       'Wage_bracket_15000_to_19999', 'Wage_bracket_20000_to_24999',
       'Wage_bracket_25000_to_29999', 'Wage_bracket_30000_to_49999',
       'Wage_bracket_50000_to_99999', 'Wage_bracket_100000_plus',
       'Total_number_in_wage_employment'],
      dtype='object')
```

[ ]:
```
dataset.shape
```

[ ]:
```
(168, 12)
```

[ ]:
```
industry_col = dataset.Industry.unique()
print(industry_col)
```

5.2.3 Reorder columns

I need to reorder columns in the dataframe such that column Wage_bracket_0_to_9999 (the working poor) is the dependent variable

[ ]:
```
cols = ['Wage_bracket_0_to_9999','Contribution_to_GDP', 'Growth_of_GDP',␣
↪'Industry',
       'Wage_bracket_10000_to_14999', 'Wage_bracket_15000_to_19999',
       'Wage_bracket_20000_to_24999', 'Wage_bracket_25000_to_29999',
       'Wage_bracket_30000_to_49999', 'Wage_bracket_50000_to_99999',
       'Wage_bracket_100000_plus', 'Total_number_in_wage_employment']
```

```
#Reorder columns
dataset = dataset.reindex(columns=cols)
dataset.columns
```

[ ]:
```
Index(['Wage_bracket_0_to_9999', 'Contribution_to_GDP', 'Growth_of_GDP',
       'Industry', 'Wage_bracket_10000_to_14999',
       'Wage_bracket_15000_to_19999', 'Wage_bracket_20000_to_24999',
       'Wage_bracket_25000_to_29999', 'Wage_bracket_30000_to_49999',
       'Wage_bracket_50000_to_99999', 'Wage_bracket_100000_plus',
       'Total_number_in_wage_employment'],
      dtype='object')
```

5.2.4 Displaying Wage_bracket columns to drop

[ ]:
```
dataset.columns[4:12]
```

[ ]:
```
Index(['Wage_bracket_10000_to_14999', 'Wage_bracket_15000_to_19999',
       'Wage_bracket_20000_to_24999', 'Wage_bracket_25000_to_29999',
       'Wage_bracket_30000_to_49999', 'Wage_bracket_50000_to_99999',
       'Wage_bracket_100000_plus', 'Total_number_in_wage_employment'],
      dtype='object')
```

[ ]:
```
# drop columns I don't want to predict
dataset.drop(dataset.columns[4:12], axis=1, inplace=True)
print(dataset)
```

```
            Wage_bracket_0_to_9999  ...
Industry
Year                                ...
2011-01-01                   12141  ...                     Agriculture, Forestry
And Fishing
2011-01-01                      59  ...                                Mining And
Quarrying
2011-01-01                     632  ...
Manufacturing
2011-01-01                    1680  ...  Electricity, Gas, Steam And Air
Conditioning S...
2011-01-01                    1680  ...  Water Supply; Sewerage, Waste
Management And R...
...                            ...  ...
...
2018-01-01                    1680  ...          Human Health And Social Work
Activities
2018-01-01                    1680  ...                Arts, Entertainment And
Recreation
2018-01-01                    1680  ...                             Other Service
Activities
2018-01-01                    1680  ...  Activities Of Households As Employers;
```

```
Undiffe...
2018-01-01                    1680  ...  Activities Of Extraterritorial
Organizations A...

[168 rows x 4 columns]
```

[ ]:
```
dataset.columns
```

[ ]:
```
Index(['Wage_bracket_0_to_9999', 'Contribution_to_GDP', 'Growth_of_GDP',
       'Industry'],
      dtype='object')
```

[ ]:
```
dataset.shape
```

[ ]:
```
(168, 4)
```

5.2.5 One-hot encoding Industry column, using Pandas get_dummies()

[ ]:
```
# generate binary values using get_dummies
dum_df = get_dummies(dataset, columns=["Industry"])# merge with main df␣
 ↪bridge_df on key values
#dataset = dataset.merge(dum_df)
#dataset
dum_df
```

[ ]:
```
            Wage_bracket_0_to_9999  ...  Industry_Wholesale And Retail Trade;
Repair Of Motor Vehicles And Motorcycles
Year                                ...
2011-01-01                   12141  ...
0
2011-01-01                      59  ...
0
2011-01-01                     632  ...
0
2011-01-01                    1680  ...
0
2011-01-01                    1680  ...
0
...                            ...  ...
...
2018-01-01                    1680  ...
0
2018-01-01                    1680  ...
0
2018-01-01                    1680  ...
0
2018-01-01                    1680  ...
0
2018-01-01                    1680  ...
```

```
0

[168 rows x 24 columns]
```

[ ]:
```
dataset=dum_df
dataset
```

[ ]:
```
            Wage_bracket_0_to_9999  ...  Industry_Wholesale And Retail Trade;
Repair Of Motor Vehicles And Motorcycles
Year                                ...
2011-01-01                   12141  ...
0
2011-01-01                      59  ...
0
2011-01-01                     632  ...
0
2011-01-01                    1680  ...
0
2011-01-01                    1680  ...
0
...                            ...  ...
...
2018-01-01                    1680  ...
0
2018-01-01                    1680  ...
0
2018-01-01                    1680  ...
0
2018-01-01                    1680  ...
0
2018-01-01                    1680  ...
0

[168 rows x 24 columns]
```

[ ]:
```
len(dataset.columns)
```

[ ]:
```
24
```

[ ]:
```
dataset.columns
```

[ ]:
```
Index(['Wage_bracket_0_to_9999', 'Contribution_to_GDP', 'Growth_of_GDP',
       'Industry_of_Accommodation And Food Service Activities',
       'Industry_of_Activities Of Extraterritorial Organizations And Bodies',
       'Industry_of_Activities Of Households As Employers; Undifferentiated
Goods- And Services-Producing Activities Of Households For Own Use',
       'Industry_of_Administrative And Support Service Activities',
       'Industry_of_Agriculture, Forestry And Fishing',
       'Industry_of_Arts, Entertainment And Recreation',
       'Industry_of_Construction', 'Industry_of_Education',
```

```
        'Industry_of_Electricity, Gas, Steam And Air Conditioning Supply',
        'Industry_of_Financial And Insurance Activities',
        'Industry_of_Human Health And Social Work Activities',
        'Industry_of_Information And Communication',
        'Industry_of_Manufacturing', 'Industry_of_Mining And Quarrying',
        'Industry_of_Other Service Activities',
        'Industry_of_Professional, Scientific And Technical Activities',
        'Industry_of_Public Administration And Defence; Compulsory Social
Security',
        'Industry_of_Real Estate Activities',
        'Industry_of_Transportation And Storage',
        'Industry_of_Water Supply; Sewerage, Waste Management And Remediation
Activities',
        'Industry_of_Wholesale And Retail Trade; Repair Of Motor Vehicles And
Motorcycles'],
      dtype='object')
```

5.2.6 Dropping Industry column

[ ]:
```
dataset.shape
```

[ ]:
```
(168, 24)
```

5.2.7 Exporting dataframe to final CSV dataset

[ ]:
```
pathFinal = '/content/drive/My Drive/ColabNotebooks/Project/'

dataset.to_csv(pathFinal + 'Wage_Employment_and_GDP_2011_to_2018_Final_1.csv') ␣
 ↪#saving final dataset
```

5.2.8 Normalizing Features

I normalize features using the MinMaxScaler

[ ]:
```
values = dataset.values
# ensure all data is float
values = values.astype('float32')
# normalize features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
```

[ ]:
```
scaled.shape
```

[ ]:
```
(168, 24)
```

5.2.9 Saving Scaler

[ ]:
```
import joblib

pathScaler = '/content/drive/My Drive/ColabNotebooks/Project/
↪wage_employment_scaler.pkl'
joblib.dump(scaler, pathScaler)
```

[ ]:
```
['/content/drive/My Drive/ColabNotebooks/Project/wage_employment_scaler.pkl']
```

5.2.10 Frame as supervised learning problem

[ ]:
```
# frame as supervised learning
reframed = series_to_supervised(scaled, 1, 1)
reframed.columns
```

[ ]:
```
Index(['var1(t-1)', 'var2(t-1)', 'var3(t-1)', 'var4(t-1)', 'var5(t-1)',
       'var6(t-1)', 'var7(t-1)', 'var8(t-1)', 'var9(t-1)', 'var10(t-1)',
       'var11(t-1)', 'var12(t-1)', 'var13(t-1)', 'var14(t-1)', 'var15(t-1)',
       'var16(t-1)', 'var17(t-1)', 'var18(t-1)', 'var19(t-1)', 'var20(t-1)',
       'var21(t-1)', 'var22(t-1)', 'var23(t-1)', 'var24(t-1)', 'var1(t)',
       'var2(t)', 'var3(t)', 'var4(t)', 'var5(t)', 'var6(t)', 'var7(t)',
       'var8(t)', 'var9(t)', 'var10(t)', 'var11(t)', 'var12(t)', 'var13(t)',
       'var14(t)', 'var15(t)', 'var16(t)', 'var17(t)', 'var18(t)', 'var19(t)',
       'var20(t)', 'var21(t)', 'var22(t)', 'var23(t)', 'var24(t)'],
      dtype='object')
```

[ ]:
```
reframed.shape
```

5.2.11 Drop columns not needed to be predicted

I need to predict Wage_bracket_0_to_9999 (the working poor)

[ ]:
```
# displaying columns to drop
reframed.columns[25:]
```

[ ]:
```
Index(['var2(t)', 'var3(t)', 'var4(t)', 'var5(t)', 'var6(t)', 'var7(t)',
       'var8(t)', 'var9(t)', 'var10(t)', 'var11(t)', 'var12(t)', 'var13(t)',
       'var14(t)', 'var15(t)', 'var16(t)', 'var17(t)', 'var18(t)', 'var19(t)',
       'var20(t)', 'var21(t)', 'var22(t)', 'var23(t)', 'var24(t)'],
      dtype='object')
```

[ ]:
```
# drop columns I don't want to predict
reframed.drop(reframed.columns[25:], axis=1, inplace=True)
print(reframed.head())
```

```
   var1(t-1)  var2(t-1)  var3(t-1)  ...  var23(t-1)  var24(t-1)   var1(t)
1   0.911882   0.707447   0.509804  ...         0.0         0.0  0.002709
2   0.002709   0.093085   0.666667  ...         0.0         0.0  0.045827
3   0.045827   0.329787   0.563025  ...         0.0         0.0  0.124690
```

```
4    0.124690    0.085106    0.344538   ...          0.0          0.0   0.124690
5    0.124690    0.093085    0.551821   ...          1.0          0.0   0.124690

[5 rows x 25 columns]
```

- The output of the above code shows that there are 32 input variables (input series); and 1 output variable for Wage_bracket_0_to_9999, at the current time in year.

[ ]:
```
reframed.shape
```

Splitting data into train and test sets

[ ]:
```
values = reframed.values

# Setting training data to be the first 7 years of data
n_train_years = 7*21

# split into train and test sets
train = values[:n_train_years, :]
test = values[n_train_years:, :]

# split into input and outputs
train_X, train_y = train[:, :-1], train[:, -1]
test_X, test_y = test[:, :-1], test[:, -1]

# reshape input to be 3D [samples, timesteps, features]
train_X = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)
```

```
(147, 1, 24) (147,) (20, 1, 24) (20,)
```

5.2.12 Design and fitting of network

I use a Vanilla LSTM model in the design of the network. A Vanilla LSTM is an LSTM model with a single hidden layer of LSTM units, and an output layer for prediction.

I define the Vanilla LSTM with 50 neurons in the first hidden layer; and 1 neuron in the output layer for predicting Wage_bracket_0_to_9999. The input shape will be 1 time step with 32 features

I adopted the Vanilla LSTM since I did not get any considerable improvement of model performance with additional layers, which comprise a Stacked LSTM.

I set the batch size to 7, since in Keras, better results are achieved when the batch size is small (32 or lower), and even best when it is a factor of the train/test sets.

I then use the Mean Absolute Error (MAE) loss function, and the optimized version of the stochastic gradient descent, known as the Adam version.

To minimize overfitting, I add a Dropout layer, with a dropout of 20%. I also use the Keras callbacks of *EarlyStopping* to halt training and thus reduce overfitting of the model, and *ModelCheckpoint* to save the best model observed during training i.e the model with least val_loss

```
[ ]: from keras.layers import Dropout

# Initialising the RNN
model = Sequential()
model.add(LSTM(units = 50, input_shape=(train_X.shape[1], train_X.shape[2])))

# Adding a dropout of 20% to minimize overfitting
model.add(Dropout(0.2))


# Adding the output layer
# For Full connection layer I use dense, with unit=1 since output is 1D
# I use softplus activation, since I want the output to be only positive values␣
 ↪(0 to +ve inf)
model.add(Dense(1, activation='softplus'))

#compiling the model
model.compile(loss='mae', optimizer='adam')

from keras.callbacks import EarlyStopping, ModelCheckpoint

pathModel = '/content/drive/My Drive/ColabNotebooks/Project/
 ↪predictive_analysis_of_Wage_bracket_0_to_9999_model.h5'
# Create callbacks -- EarlyStopping, ModelCheckpoint

# EarlyStopping callback with patience
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=20)

# ModelCheckpoint callback, for saving best model
mc = ModelCheckpoint(pathModel, monitor='val_loss', mode='min', verbose=1,␣
 ↪save_best_only=True)

###############################################-----------############################
# fit network
history = model.fit(train_X, train_y, epochs= 200, batch_size=7,
                    validation_data=(test_X, test_y), verbose=2, shuffle=False,
                    callbacks=[es, mc])
```

```
Epoch 1/200
21/21 - 3s - loss: 0.5480 - val_loss: 0.5123

Epoch 00001: val_loss improved from inf to 0.51231, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 2/200
21/21 - 0s - loss: 0.4823 - val_loss: 0.4407
```

```
Epoch 00002: val_loss improved from 0.51231 to 0.44072, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 3/200
21/21 - 0s - loss: 0.4149 - val_loss: 0.3601

Epoch 00003: val_loss improved from 0.44072 to 0.36006, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 4/200
21/21 - 0s - loss: 0.3376 - val_loss: 0.2721

Epoch 00004: val_loss improved from 0.36006 to 0.27213, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 5/200
21/21 - 0s - loss: 0.2558 - val_loss: 0.1855

Epoch 00005: val_loss improved from 0.27213 to 0.18549, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 6/200
21/21 - 0s - loss: 0.1846 - val_loss: 0.1096

Epoch 00006: val_loss improved from 0.18549 to 0.10959, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 7/200
21/21 - 0s - loss: 0.1249 - val_loss: 0.0598

Epoch 00007: val_loss improved from 0.10959 to 0.05984, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 8/200
21/21 - 0s - loss: 0.0841 - val_loss: 0.0364

Epoch 00008: val_loss improved from 0.05984 to 0.03641, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 9/200
21/21 - 0s - loss: 0.0704 - val_loss: 0.0349

Epoch 00009: val_loss improved from 0.03641 to 0.03495, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 10/200
21/21 - 0s - loss: 0.0679 - val_loss: 0.0342
```

```
Epoch 00010: val_loss improved from 0.03495 to 0.03421, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 11/200
21/21 - 0s - loss: 0.0635 - val_loss: 0.0321

Epoch 00011: val_loss improved from 0.03421 to 0.03211, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 12/200
21/21 - 0s - loss: 0.0592 - val_loss: 0.0316

Epoch 00012: val_loss improved from 0.03211 to 0.03158, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 13/200
21/21 - 0s - loss: 0.0584 - val_loss: 0.0309

Epoch 00013: val_loss improved from 0.03158 to 0.03091, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 14/200
21/21 - 0s - loss: 0.0568 - val_loss: 0.0285

Epoch 00014: val_loss improved from 0.03091 to 0.02849, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 15/200
21/21 - 0s - loss: 0.0527 - val_loss: 0.0260

Epoch 00015: val_loss improved from 0.02849 to 0.02605, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 16/200
21/21 - 0s - loss: 0.0496 - val_loss: 0.0242

Epoch 00016: val_loss improved from 0.02605 to 0.02424, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 17/200
21/21 - 0s - loss: 0.0480 - val_loss: 0.0236

Epoch 00017: val_loss improved from 0.02424 to 0.02356, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 18/200
21/21 - 0s - loss: 0.0468 - val_loss: 0.0224
```

```
Epoch 00018: val_loss improved from 0.02356 to 0.02241, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 19/200
21/21 - 0s - loss: 0.0412 - val_loss: 0.0220

Epoch 00019: val_loss improved from 0.02241 to 0.02200, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 20/200
21/21 - 0s - loss: 0.0424 - val_loss: 0.0223

Epoch 00020: val_loss did not improve from 0.02200
Epoch 21/200
21/21 - 0s - loss: 0.0376 - val_loss: 0.0206

Epoch 00021: val_loss improved from 0.02200 to 0.02057, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 22/200
21/21 - 0s - loss: 0.0360 - val_loss: 0.0191

Epoch 00022: val_loss improved from 0.02057 to 0.01914, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 23/200
21/21 - 0s - loss: 0.0333 - val_loss: 0.0192

Epoch 00023: val_loss did not improve from 0.01914
Epoch 24/200
21/21 - 0s - loss: 0.0330 - val_loss: 0.0208

Epoch 00024: val_loss did not improve from 0.01914
Epoch 25/200
21/21 - 0s - loss: 0.0309 - val_loss: 0.0211

Epoch 00025: val_loss did not improve from 0.01914
Epoch 26/200
21/21 - 0s - loss: 0.0269 - val_loss: 0.0186

Epoch 00026: val_loss improved from 0.01914 to 0.01862, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 27/200
21/21 - 0s - loss: 0.0284 - val_loss: 0.0176

Epoch 00027: val_loss improved from 0.01862 to 0.01762, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
```

```
et_0_to_9999_model.h5
Epoch 28/200
21/21 - 0s - loss: 0.0275 - val_loss: 0.0186

Epoch 00028: val_loss did not improve from 0.01762
Epoch 29/200
21/21 - 0s - loss: 0.0251 - val_loss: 0.0189

Epoch 00029: val_loss did not improve from 0.01762
Epoch 30/200
21/21 - 0s - loss: 0.0253 - val_loss: 0.0183

Epoch 00030: val_loss did not improve from 0.01762
Epoch 31/200
21/21 - 0s - loss: 0.0261 - val_loss: 0.0185

Epoch 00031: val_loss did not improve from 0.01762
Epoch 32/200
21/21 - 0s - loss: 0.0262 - val_loss: 0.0180

Epoch 00032: val_loss did not improve from 0.01762
Epoch 33/200
21/21 - 0s - loss: 0.0251 - val_loss: 0.0179

Epoch 00033: val_loss did not improve from 0.01762
Epoch 34/200
21/21 - 0s - loss: 0.0249 - val_loss: 0.0162

Epoch 00034: val_loss improved from 0.01762 to 0.01620, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
et_0_to_9999_model.h5
Epoch 35/200
21/21 - 0s - loss: 0.0259 - val_loss: 0.0164

Epoch 00035: val_loss did not improve from 0.01620
Epoch 36/200
21/21 - 0s - loss: 0.0251 - val_loss: 0.0191

Epoch 00036: val_loss did not improve from 0.01620
Epoch 37/200
21/21 - 0s - loss: 0.0247 - val_loss: 0.0186

Epoch 00037: val_loss did not improve from 0.01620
Epoch 38/200
21/21 - 0s - loss: 0.0244 - val_loss: 0.0160

Epoch 00038: val_loss improved from 0.01620 to 0.01600, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Wage_brack
```

```
et_0_to_9999_model.h5
Epoch 39/200
21/21 - 0s - loss: 0.0262 - val_loss: 0.0172

Epoch 00039: val_loss did not improve from 0.01600
Epoch 40/200
21/21 - 0s - loss: 0.0236 - val_loss: 0.0180

Epoch 00040: val_loss did not improve from 0.01600
Epoch 41/200
21/21 - 0s - loss: 0.0240 - val_loss: 0.0183

Epoch 00041: val_loss did not improve from 0.01600
Epoch 42/200
21/21 - 0s - loss: 0.0260 - val_loss: 0.0202

Epoch 00042: val_loss did not improve from 0.01600
Epoch 43/200
21/21 - 0s - loss: 0.0265 - val_loss: 0.0192

Epoch 00043: val_loss did not improve from 0.01600
Epoch 44/200
21/21 - 0s - loss: 0.0214 - val_loss: 0.0191

Epoch 00044: val_loss did not improve from 0.01600
Epoch 45/200
21/21 - 0s - loss: 0.0227 - val_loss: 0.0177

Epoch 00045: val_loss did not improve from 0.01600
Epoch 46/200
21/21 - 0s - loss: 0.0234 - val_loss: 0.0184

Epoch 00046: val_loss did not improve from 0.01600
Epoch 47/200
21/21 - 0s - loss: 0.0235 - val_loss: 0.0193

Epoch 00047: val_loss did not improve from 0.01600
Epoch 48/200
21/21 - 0s - loss: 0.0242 - val_loss: 0.0182

Epoch 00048: val_loss did not improve from 0.01600
Epoch 49/200
21/21 - 0s - loss: 0.0247 - val_loss: 0.0168

Epoch 00049: val_loss did not improve from 0.01600
Epoch 50/200
21/21 - 0s - loss: 0.0246 - val_loss: 0.0173
```

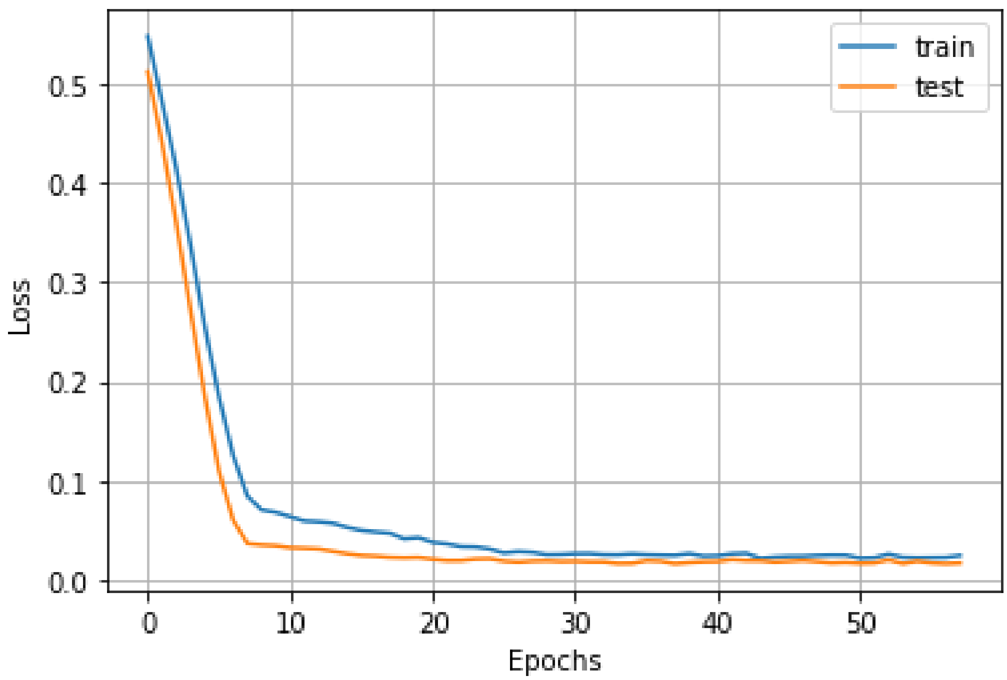
```
Epoch 00050: val_loss did not improve from 0.01600
Epoch 51/200
21/21 - 0s - loss: 0.0216 - val_loss: 0.0164

Epoch 00051: val_loss did not improve from 0.01600
Epoch 52/200
21/21 - 0s - loss: 0.0220 - val_loss: 0.0168

Epoch 00052: val_loss did not improve from 0.01600
Epoch 53/200
21/21 - 0s - loss: 0.0256 - val_loss: 0.0199

Epoch 00053: val_loss did not improve from 0.01600
Epoch 54/200
21/21 - 0s - loss: 0.0224 - val_loss: 0.0162

Epoch 00054: val_loss did not improve from 0.01600
Epoch 55/200
21/21 - 0s - loss: 0.0220 - val_loss: 0.0187

Epoch 00055: val_loss did not improve from 0.01600
Epoch 56/200
21/21 - 0s - loss: 0.0222 - val_loss: 0.0168

Epoch 00056: val_loss did not improve from 0.01600
Epoch 57/200
21/21 - 0s - loss: 0.0223 - val_loss: 0.0160

Epoch 00057: val_loss did not improve from 0.01600
Epoch 58/200
21/21 - 0s - loss: 0.0242 - val_loss: 0.0167

Epoch 00058: val_loss did not improve from 0.01600
Epoch 00058: early stopping
```

5.2.13 Plotting loss charts

Use the history object to get the saved performance results

[ ]:
```
import matplotlib.pyplot as plt

# plot history
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.grid(True)
plt.legend()
```

```
plt.show()
```



[ ]:

[ ]:

5.2.14 ———————————————————————— End of Modelling Model 1 ———————————————————————————

[ ]:

[ ]:

6 Part 6: Model Evaluation

6.1 Workflow

1. Load saved model
2. Perform a prediction
3. Evaluate model using metrics(RMSE) and curves

6.1.1 Lodaing saved model

[ ]:
```
#load model
from tensorflow.keras.models import load_model
pathModel = '/content/drive/My Drive/ColabNotebooks/Project/
↪predictive_analysis_of_Wage_bracket_0_to_9999_model.h5'

model = load_model(pathModel)
```

6.1.2 Making a prediction

[ ]:
```
# make a prediction
yhat = model.predict(test_X)

#reshaping test_X
test_x = test_X.reshape((test_X.shape[0], test_X.shape[2]))

# invert scaling for forecast
inv_yhat = concatenate((yhat, test_x[:, 1:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:,0]

# invert scaling for actual
test_y = test_y.reshape((len(test_y), 1))
inv_y = concatenate((test_y, test_x[:, 1:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:,0]
```

[ ]:
```
```

[ ]: 20

6.1.3 Displaying Predicted Values: Number of people in Wage_bracket_0_to_9999 (Working Poor)

[ ]:
```
# Type casting predictions to int
inv_yhat = inv_yhat.astype(int)

# Displaying Predicted values
print('Predicted values are: ')
print(inv_yhat)
```

```
Predicted values are:
[ 142  962 1700 1593 1688  392  832  310  367 1492  333 3027  295 1382
  872 1580 1669 1662 1638 1715]
```

6.1.4 Evaluating model using performance metrics: Root Mean Square Error (RMSE)

The RMSE is used to measure the error of a model in predicting quantitave data, i.e the differences between the predicted and actual values. The RMSE is given in units of the dependent variable (in this case units of Wage_bracket_0_to_9999, which is the number of people in that wage bracket)

[ ]:
```
# calculating RMSE
rmse = sqrt(mean_squared_error(inv_y, inv_yhat))
print('Root Mean Square Error: %.3f' % rmse)

#Normalizing RMSE using range
nrmse = rmse/ (inv_y.max() - inv_yhat.min())
print('Normalized Root Mean Square Error: %.3f' %nrmse)

#Accuracy of model
accuracy = 1-nrmse
print("Accuracy of model is: %.3f " % accuracy)
```

```
Root Mean Square Error: 398.440
Normalized Root Mean Square Error: 0.137
Accuracy of model is: 0.863
```

The RMSE is used as a heuristic for training models, and to evaluate trained models for usefulness/accuracy. In the first case, I was able to tweak the model hyperparameters to ensure I decrease the RMSE with each tuning of the model hyperparameters.

In the second case, I was able to tune the model hyperparameters to record a low RMSE of 403.810, which was achieved through a vanilla LSTM with a batch size of 7, which was trained for 100 epochs.

Normalizing the RMSE in the range (0,1) yields an NRMSE of 0.037, which is considerably low, hence showing the model perfomed very well. With subtracting the NRMSE from 1, I can loosely say the model has an accuracy of about 96.3 %

6.1.5 Plotting Actual vs Forecast values

[ ]:
```
import matplotlib.pyplot as plt
plt.plot(inv_y, color = 'black', label = 'Actual')
plt.plot(inv_yhat, color = 'Green', label = 'Predicted')
plt.title('Wage Employment')
plt.xlabel('Industry Columns')
plt.ylabel('Wage_bracket_0_to_9999')
plt.legend()
plt.show()
```

The plots of the actual vs the forecast values are closely knit, depicting that the model was able to capture the trend of the number of people earning below USD 2 a day (working poor -- Wage_bracket_0_to_9999) according to their Industry/Sector of employment, based on the input at the prior time step

[ ]:

```
```

[ ]:

```
```

7 -------------------------- Model 2: Predictive Analysis of Total_employment ---------------------------------

7.1 Defination of Problem

I define the supervised learning problem as predicting the total number of people in wage employment (*Total_number_in_wage_employment*) at the current year (t), given the GDP indicators and other inputs at the prior time step

7.1.1 Load dataset

[ ]:

```
#import pandas as pd

# load dataset
pathDataset = '/content/drive/My Drive/ColabNotebooks/Project/
↪Wage_Employment_and_GDP_2011_to_2018_Updated.csv'
```

```
dataset = read_csv(pathDataset, header=0, index_col=0)
#dataset.drop('Industry', axis=1, inplace=True)
dataset.drop('Unnamed: 0', axis=1, inplace=True)
print(dataset.columns)
```

```
Index(['Industry', 'Contribution_to_GDP', 'Growth_of_GDP',
       'Wage_bracket_0_to_9999', 'Wage_bracket_10000_to_14999',
       'Wage_bracket_15000_to_19999', 'Wage_bracket_20000_to_24999',
       'Wage_bracket_25000_to_29999', 'Wage_bracket_30000_to_49999',
       'Wage_bracket_50000_to_99999', 'Wage_bracket_100000_plus',
       'Total_number_in_wage_employment'],
      dtype='object')
```

7.1.2 Reorder columns

I reorder the columns in the dataframe such that the Total_number_in_wage_employment column is now the dependent variable

[ ]:
```
cols = ['Total_number_in_wage_employment', 'Contribution_to_GDP',
        'Growth_of_GDP', 'Industry', 'Wage_bracket_0_to_9999',
       'Wage_bracket_10000_to_14999', 'Wage_bracket_15000_to_19999',
       'Wage_bracket_20000_to_24999', 'Wage_bracket_25000_to_29999',
       'Wage_bracket_30000_to_49999', 'Wage_bracket_50000_to_99999',
       'Wage_bracket_100000_plus']
#Reorder columns
dataset = dataset.reindex(columns=cols)
dataset.columns
```

[ ]:
```
Index(['Total_number_in_wage_employment', 'Contribution_to_GDP',
       'Growth_of_GDP', 'Industry', 'Wage_bracket_0_to_9999',
       'Wage_bracket_10000_to_14999', 'Wage_bracket_15000_to_19999',
       'Wage_bracket_20000_to_24999', 'Wage_bracket_25000_to_29999',
       'Wage_bracket_30000_to_49999', 'Wage_bracket_50000_to_99999',
       'Wage_bracket_100000_plus'],
      dtype='object')
```

7.1.3 Displaying columns to drop

[ ]:
```
dataset.columns[4:12]
```

[ ]:
```
Index(['Wage_bracket_0_to_9999', 'Wage_bracket_10000_to_14999',
       'Wage_bracket_15000_to_19999', 'Wage_bracket_20000_to_24999',
       'Wage_bracket_25000_to_29999', 'Wage_bracket_30000_to_49999',
       'Wage_bracket_50000_to_99999', 'Wage_bracket_100000_plus'],
      dtype='object')
```

[ ]:
```
# drop columns I don't want to use in prediction (Input Series)
dataset.drop(dataset.columns[4:12], axis=1, inplace=True)
print(dataset)
```

```
            Total_number_in_wage_employment  ...
Industry
Year                                         ...
2011-01-01                           341422  ...                    Agriculture,
Forestry And Fishing
2011-01-01                             8732  ...
Mining And Quarrying
2011-01-01                           276885  ...
Manufacturing
2011-01-01                            14018  ...  Electricity, Gas, Steam And
Air Conditioning S...
2011-01-01                            21211  ...  Water Supply; Sewerage, Waste
Management And R...
...                                     ...  ...
...
2018-01-01                           150434  ...              Human Health And
Social Work Activities
2018-01-01                             8922  ...                         Arts,
Entertainment And Recreation
2018-01-01                            38012  ...                            Other
Service Activities
2018-01-01                           121703  ...  Activities Of Households As
Employers; Undiffe...
2018-01-01                           118655  ...  Activities Of Extraterritorial
Organizations A...

[168 rows x 4 columns]
```

[ ]:
```
dataset.columns
```

[ ]:
```
Index(['Total_number_in_wage_employment', 'Contribution_to_GDP',
       'Growth_of_GDP', 'Industry'],
      dtype='object')
```

7.1.4 Perform one-hot encoding of Industry column, using Pandas get_dummies() function

[ ]:
```
# generate binary values using get_dummies
dum_df = get_dummies(dataset, columns=["Industry"])# merge with main df␣
↪bridge_df on key values

# assigning dataframe with one hot encoded Industry column
dataset = dum_df
dataset
```

[ ]:
```
            Total_number_in_wage_employment  ...  Industry_Wholesale And Retail
Trade; Repair Of Motor Vehicles And Motorcycles
Year                                         ...
2011-01-01                           341422  ...
```

```
0
2011-01-01                                8732  ...
0
2011-01-01                              276885  ...
0
2011-01-01                               14018  ...
0
2011-01-01                               21211  ...
0
...                                        ...  ...
...
2018-01-01                              150434  ...
0
2018-01-01                                8922  ...
0
2018-01-01                               38012  ...
0
2018-01-01                              121703  ...
0
2018-01-01                              118655  ...
0

[168 rows x 24 columns]
```

[ ]:
```
dataset.shape
```

[ ]:
```
(168, 24)
```

7.1.5 Exporting dataframe to a final CSV dataset

[ ]:
```

```

7.1.6 Normalizing features, framing as supervised learning

[ ]:
```
values = dataset.values
# ensure all data is float
values = values.astype('float32')
# normalize features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)

# frame as supervised learning
reframed = series_to_supervised(scaled, 1, 1)
reframed.columns
```

[ ]:
```
Index(['var1(t-1)', 'var2(t-1)', 'var3(t-1)', 'var4(t-1)', 'var5(t-1)',
       'var6(t-1)', 'var7(t-1)', 'var8(t-1)', 'var9(t-1)', 'var10(t-1)',
```

```
         'var11(t-1)', 'var12(t-1)', 'var13(t-1)', 'var14(t-1)', 'var15(t-1)',
         'var16(t-1)', 'var17(t-1)', 'var18(t-1)', 'var19(t-1)', 'var20(t-1)',
         'var21(t-1)', 'var22(t-1)', 'var23(t-1)', 'var24(t-1)', 'var1(t)',
         'var2(t)', 'var3(t)', 'var4(t)', 'var5(t)', 'var6(t)', 'var7(t)',
         'var8(t)', 'var9(t)', 'var10(t)', 'var11(t)', 'var12(t)', 'var13(t)',
         'var14(t)', 'var15(t)', 'var16(t)', 'var17(t)', 'var18(t)', 'var19(t)',
         'var20(t)', 'var21(t)', 'var22(t)', 'var23(t)', 'var24(t)'],
        dtype='object')
```

7.1.7 Saving Scaler to Disk

[ ]:
```
import joblib
#path to scaler
pathScaler = '/content/drive/My Drive/ColabNotebooks/Project/
 ↪total_employment_scaler.pkl'

joblib.dump(scaler, pathScaler)
```

[ ]:
```
['/content/drive/My Drive/ColabNotebooks/Project/total_employment_scaler.pkl']
```

7.1.8 Dropping columns not intended to be predicted

I need to predict var1(t) -- Total_number_in_wage_employment

[ ]:
```
# displaying columns to drop
reframed.columns[25:]
```

[ ]:
```
Index(['var2(t)', 'var3(t)', 'var4(t)', 'var5(t)', 'var6(t)', 'var7(t)',
       'var8(t)', 'var9(t)', 'var10(t)', 'var11(t)', 'var12(t)', 'var13(t)',
       'var14(t)', 'var15(t)', 'var16(t)', 'var17(t)', 'var18(t)', 'var19(t)',
       'var20(t)', 'var21(t)', 'var22(t)', 'var23(t)', 'var24(t)'],
      dtype='object')
```

[ ]:
```
# drop columns I don't want to predict
reframed.drop(reframed.columns[25:], axis=1, inplace=True)
print(reframed.head())
```

```
   var1(t-1)  var2(t-1)  var3(t-1)  ...  var23(t-1)  var24(t-1)   var1(t)
1   0.590239   0.707447   0.509804  ...         0.0         0.0  0.011147
2   0.011147   0.093085   0.666667  ...         0.0         0.0  0.477904
3   0.477904   0.329787   0.563025  ...         0.0         0.0  0.020348
4   0.020348   0.085106   0.344538  ...         0.0         0.0  0.032868
5   0.032868   0.093085   0.551821  ...         1.0         0.0  0.185546

[5 rows x 25 columns]
```

[ ]:
```
reframed.shape
```

[ ]:
```
(167, 25)
```

7.1.9 Splitting data into train and test sets

[ ]:
```
values = reframed.values

# Setting training data to be the first 7 years of data
n_train_years = 7*21

# split into train and test sets
train = values[:n_train_years, :]
test = values[n_train_years:, :]

# split into input and outputs
train_X, train_y = train[:, :-1], train[:, -1]
test_X, test_y = test[:, :-1], test[:, -1]

# reshape input to be 3D [samples, timesteps, features]
train_X = train_X.reshape((train_X.shape[0], 1, train_X.shape[1]))
test_X = test_X.reshape((test_X.shape[0], 1, test_X.shape[1]))
print(train_X.shape, train_y.shape, test_X.shape, test_y.shape)
```

```
(147, 1, 24) (147,) (20, 1, 24) (20,)
```

7.1.10 Model Design and Fitting

[ ]:
```
from keras.layers import Dropout

# Initialising the RNN
model = Sequential()
model.add(LSTM(units = 50, input_shape=(train_X.shape[1], train_X.shape[2])))

# Adding a dropout of 20% to minimize overfitting
model.add(Dropout(0.2))

# Adding the output layer
# For Full connection layer I use dense, with unit=1 since output is 1D
# I use softplus activation, since I want the output to be only positive values␣
 ↪(0 to +ve inf)
model.add(Dense(1, activation='softplus'))

#compiling the model
model.compile(loss='mae', optimizer='adam')

from keras.callbacks import EarlyStopping, ModelCheckpoint

pathModel = '/content/drive/My Drive/ColabNotebooks/Project/
 ↪predictive_analysis_of_Total_number_in_wage_employment_model.h5'
# Create callbacks -- EarlyStopping, ModelCheckpoint
```

```
# EarlyStopping callback with patience
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=20)

# ModelCheckpoint callback, for saving best model
mc = ModelCheckpoint(pathModel, monitor='val_loss', mode='min', verbose=1,␣
 ↪save_best_only=True)

###############################################-----------#############################
# fit network
history = model.fit(train_X, train_y, epochs= 200, batch_size=7,
                    validation_data=(test_X, test_y), verbose=2, shuffle=False,
                    callbacks=[es, mc])
```

```
Epoch 1/200
21/21 - 2s - loss: 0.4922 - val_loss: 0.4560

Epoch 00001: val_loss improved from inf to 0.45604, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 2/200
21/21 - 0s - loss: 0.4335 - val_loss: 0.3945

Epoch 00002: val_loss improved from 0.45604 to 0.39455, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 3/200
21/21 - 0s - loss: 0.3691 - val_loss: 0.3310

Epoch 00003: val_loss improved from 0.39455 to 0.33097, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 4/200
21/21 - 0s - loss: 0.3111 - val_loss: 0.2743

Epoch 00004: val_loss improved from 0.33097 to 0.27434, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 5/200
21/21 - 0s - loss: 0.2569 - val_loss: 0.2232

Epoch 00005: val_loss improved from 0.27434 to 0.22324, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 6/200
21/21 - 0s - loss: 0.2063 - val_loss: 0.1798
```

```
Epoch 00006: val_loss improved from 0.22324 to 0.17984, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 7/200
21/21 - 0s - loss: 0.1677 - val_loss: 0.1511

Epoch 00007: val_loss improved from 0.17984 to 0.15109, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 8/200
21/21 - 0s - loss: 0.1414 - val_loss: 0.1365

Epoch 00008: val_loss improved from 0.15109 to 0.13650, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 9/200
21/21 - 0s - loss: 0.1291 - val_loss: 0.1326

Epoch 00009: val_loss improved from 0.13650 to 0.13260, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 10/200
21/21 - 0s - loss: 0.1151 - val_loss: 0.1276

Epoch 00010: val_loss improved from 0.13260 to 0.12764, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 11/200
21/21 - 0s - loss: 0.1055 - val_loss: 0.1219

Epoch 00011: val_loss improved from 0.12764 to 0.12191, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 12/200
21/21 - 0s - loss: 0.1013 - val_loss: 0.1135

Epoch 00012: val_loss improved from 0.12191 to 0.11348, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 13/200
21/21 - 0s - loss: 0.0927 - val_loss: 0.1055

Epoch 00013: val_loss improved from 0.11348 to 0.10551, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 14/200
21/21 - 0s - loss: 0.0822 - val_loss: 0.0980
```

```
Epoch 00014: val_loss improved from 0.10551 to 0.09800, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 15/200
21/21 - 0s - loss: 0.0721 - val_loss: 0.0865

Epoch 00015: val_loss improved from 0.09800 to 0.08646, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 16/200
21/21 - 0s - loss: 0.0634 - val_loss: 0.0806

Epoch 00016: val_loss improved from 0.08646 to 0.08058, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 17/200
21/21 - 0s - loss: 0.0544 - val_loss: 0.0732

Epoch 00017: val_loss improved from 0.08058 to 0.07318, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 18/200
21/21 - 0s - loss: 0.0458 - val_loss: 0.0674

Epoch 00018: val_loss improved from 0.07318 to 0.06745, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 19/200
21/21 - 0s - loss: 0.0486 - val_loss: 0.0642

Epoch 00019: val_loss improved from 0.06745 to 0.06424, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 20/200
21/21 - 0s - loss: 0.0448 - val_loss: 0.0594

Epoch 00020: val_loss improved from 0.06424 to 0.05936, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 21/200
21/21 - 0s - loss: 0.0426 - val_loss: 0.0569

Epoch 00021: val_loss improved from 0.05936 to 0.05693, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 22/200
21/21 - 0s - loss: 0.0413 - val_loss: 0.0569
```

```
Epoch 00022: val_loss improved from 0.05693 to 0.05687, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 23/200
21/21 - 0s - loss: 0.0368 - val_loss: 0.0550

Epoch 00023: val_loss improved from 0.05687 to 0.05498, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 24/200
21/21 - 0s - loss: 0.0378 - val_loss: 0.0536

Epoch 00024: val_loss improved from 0.05498 to 0.05361, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 25/200
21/21 - 0s - loss: 0.0355 - val_loss: 0.0516

Epoch 00025: val_loss improved from 0.05361 to 0.05157, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 26/200
21/21 - 0s - loss: 0.0363 - val_loss: 0.0529

Epoch 00026: val_loss did not improve from 0.05157
Epoch 27/200
21/21 - 0s - loss: 0.0339 - val_loss: 0.0564

Epoch 00027: val_loss did not improve from 0.05157
Epoch 28/200
21/21 - 0s - loss: 0.0359 - val_loss: 0.0553

Epoch 00028: val_loss did not improve from 0.05157
Epoch 29/200
21/21 - 0s - loss: 0.0316 - val_loss: 0.0522

Epoch 00029: val_loss did not improve from 0.05157
Epoch 30/200
21/21 - 0s - loss: 0.0331 - val_loss: 0.0521

Epoch 00030: val_loss did not improve from 0.05157
Epoch 31/200
21/21 - 0s - loss: 0.0328 - val_loss: 0.0505

Epoch 00031: val_loss improved from 0.05157 to 0.05050, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 32/200
```

```
21/21 - 0s - loss: 0.0303 - val_loss: 0.0490

Epoch 00032: val_loss improved from 0.05050 to 0.04903, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 33/200
21/21 - 0s - loss: 0.0289 - val_loss: 0.0503

Epoch 00033: val_loss did not improve from 0.04903
Epoch 34/200
21/21 - 0s - loss: 0.0310 - val_loss: 0.0530

Epoch 00034: val_loss did not improve from 0.04903
Epoch 35/200
21/21 - 0s - loss: 0.0305 - val_loss: 0.0523

Epoch 00035: val_loss did not improve from 0.04903
Epoch 36/200
21/21 - 0s - loss: 0.0309 - val_loss: 0.0506

Epoch 00036: val_loss did not improve from 0.04903
Epoch 37/200
21/21 - 0s - loss: 0.0289 - val_loss: 0.0488

Epoch 00037: val_loss improved from 0.04903 to 0.04879, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 38/200
21/21 - 0s - loss: 0.0282 - val_loss: 0.0467

Epoch 00038: val_loss improved from 0.04879 to 0.04667, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 39/200
21/21 - 0s - loss: 0.0306 - val_loss: 0.0490

Epoch 00039: val_loss did not improve from 0.04667
Epoch 40/200
21/21 - 0s - loss: 0.0289 - val_loss: 0.0479

Epoch 00040: val_loss did not improve from 0.04667
Epoch 41/200
21/21 - 0s - loss: 0.0275 - val_loss: 0.0457

Epoch 00041: val_loss improved from 0.04667 to 0.04568, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 42/200
```

```
21/21 - 0s - loss: 0.0295 - val_loss: 0.0504

Epoch 00042: val_loss did not improve from 0.04568
Epoch 43/200
21/21 - 0s - loss: 0.0282 - val_loss: 0.0470

Epoch 00043: val_loss did not improve from 0.04568
Epoch 44/200
21/21 - 0s - loss: 0.0284 - val_loss: 0.0492

Epoch 00044: val_loss did not improve from 0.04568
Epoch 45/200
21/21 - 0s - loss: 0.0301 - val_loss: 0.0496

Epoch 00045: val_loss did not improve from 0.04568
Epoch 46/200
21/21 - 0s - loss: 0.0256 - val_loss: 0.0424

Epoch 00046: val_loss improved from 0.04568 to 0.04242, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 47/200
21/21 - 0s - loss: 0.0279 - val_loss: 0.0453

Epoch 00047: val_loss did not improve from 0.04242
Epoch 48/200
21/21 - 0s - loss: 0.0268 - val_loss: 0.0424

Epoch 00048: val_loss improved from 0.04242 to 0.04238, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 49/200
21/21 - 0s - loss: 0.0256 - val_loss: 0.0447

Epoch 00049: val_loss did not improve from 0.04238
Epoch 50/200
21/21 - 0s - loss: 0.0283 - val_loss: 0.0464

Epoch 00050: val_loss did not improve from 0.04238
Epoch 51/200
21/21 - 0s - loss: 0.0269 - val_loss: 0.0473

Epoch 00051: val_loss did not improve from 0.04238
Epoch 52/200
21/21 - 0s - loss: 0.0285 - val_loss: 0.0440

Epoch 00052: val_loss did not improve from 0.04238
Epoch 53/200
```

```
21/21 - 0s - loss: 0.0276 - val_loss: 0.0454

Epoch 00053: val_loss did not improve from 0.04238
Epoch 54/200
21/21 - 0s - loss: 0.0262 - val_loss: 0.0454

Epoch 00054: val_loss did not improve from 0.04238
Epoch 55/200
21/21 - 0s - loss: 0.0245 - val_loss: 0.0391

Epoch 00055: val_loss improved from 0.04238 to 0.03910, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 56/200
21/21 - 0s - loss: 0.0277 - val_loss: 0.0440

Epoch 00056: val_loss did not improve from 0.03910
Epoch 57/200
21/21 - 0s - loss: 0.0284 - val_loss: 0.0479

Epoch 00057: val_loss did not improve from 0.03910
Epoch 58/200
21/21 - 0s - loss: 0.0231 - val_loss: 0.0454

Epoch 00058: val_loss did not improve from 0.03910
Epoch 59/200
21/21 - 0s - loss: 0.0268 - val_loss: 0.0431

Epoch 00059: val_loss did not improve from 0.03910
Epoch 60/200
21/21 - 0s - loss: 0.0274 - val_loss: 0.0451

Epoch 00060: val_loss did not improve from 0.03910
Epoch 61/200
21/21 - 0s - loss: 0.0270 - val_loss: 0.0407

Epoch 00061: val_loss did not improve from 0.03910
Epoch 62/200
21/21 - 0s - loss: 0.0274 - val_loss: 0.0456

Epoch 00062: val_loss did not improve from 0.03910
Epoch 63/200
21/21 - 0s - loss: 0.0242 - val_loss: 0.0367

Epoch 00063: val_loss improved from 0.03910 to 0.03670, saving model to
/content/drive/My Drive/ColabNotebooks/Project/predictive_analysis_of_Total_numb
er_in_wage_employment_model.h5
Epoch 64/200
```

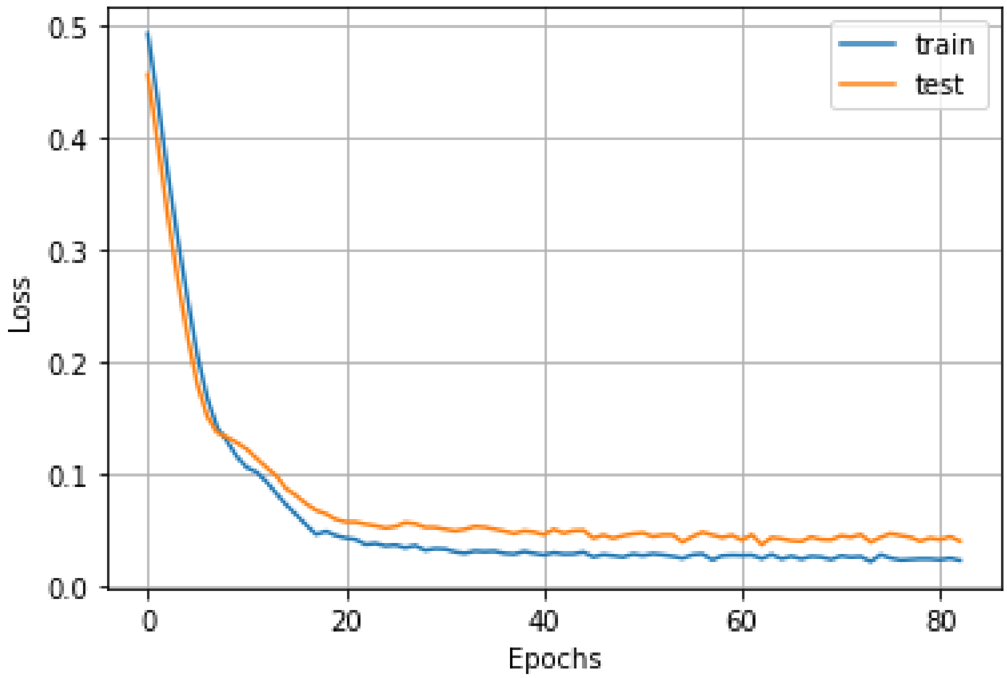
```
21/21 - 0s - loss: 0.0280 - val_loss: 0.0431

Epoch 00064: val_loss did not improve from 0.03670
Epoch 65/200
21/21 - 0s - loss: 0.0240 - val_loss: 0.0420

Epoch 00065: val_loss did not improve from 0.03670
Epoch 66/200
21/21 - 0s - loss: 0.0264 - val_loss: 0.0402

Epoch 00066: val_loss did not improve from 0.03670
Epoch 67/200
21/21 - 0s - loss: 0.0239 - val_loss: 0.0397

Epoch 00067: val_loss did not improve from 0.03670
Epoch 68/200
21/21 - 0s - loss: 0.0263 - val_loss: 0.0436

Epoch 00068: val_loss did not improve from 0.03670
Epoch 69/200
21/21 - 0s - loss: 0.0256 - val_loss: 0.0413

Epoch 00069: val_loss did not improve from 0.03670
Epoch 70/200
21/21 - 0s - loss: 0.0234 - val_loss: 0.0409

Epoch 00070: val_loss did not improve from 0.03670
Epoch 71/200
21/21 - 0s - loss: 0.0267 - val_loss: 0.0447

Epoch 00071: val_loss did not improve from 0.03670
Epoch 72/200
21/21 - 0s - loss: 0.0255 - val_loss: 0.0434

Epoch 00072: val_loss did not improve from 0.03670
Epoch 73/200
21/21 - 0s - loss: 0.0261 - val_loss: 0.0458

Epoch 00073: val_loss did not improve from 0.03670
Epoch 74/200
21/21 - 0s - loss: 0.0215 - val_loss: 0.0390

Epoch 00074: val_loss did not improve from 0.03670
Epoch 75/200
21/21 - 0s - loss: 0.0277 - val_loss: 0.0432

Epoch 00075: val_loss did not improve from 0.03670
Epoch 76/200
```

```
21/21 - 0s - loss: 0.0247 - val_loss: 0.0466

Epoch 00076: val_loss did not improve from 0.03670
Epoch 77/200
21/21 - 0s - loss: 0.0229 - val_loss: 0.0449

Epoch 00077: val_loss did not improve from 0.03670
Epoch 78/200
21/21 - 0s - loss: 0.0235 - val_loss: 0.0433

Epoch 00078: val_loss did not improve from 0.03670
Epoch 79/200
21/21 - 0s - loss: 0.0240 - val_loss: 0.0398

Epoch 00079: val_loss did not improve from 0.03670
Epoch 80/200
21/21 - 0s - loss: 0.0239 - val_loss: 0.0427

Epoch 00080: val_loss did not improve from 0.03670
Epoch 81/200
21/21 - 0s - loss: 0.0232 - val_loss: 0.0412

Epoch 00081: val_loss did not improve from 0.03670
Epoch 82/200
21/21 - 0s - loss: 0.0246 - val_loss: 0.0439

Epoch 00082: val_loss did not improve from 0.03670
Epoch 83/200
21/21 - 0s - loss: 0.0227 - val_loss: 0.0399

Epoch 00083: val_loss did not improve from 0.03670
Epoch 00083: early stopping
```

7.1.11 Plotting loss charts

Use the history object to get the saved performance results

[ ]:
```python
import matplotlib.pyplot as plt

# plot history
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.grid(True)
plt.legend()
plt.show()
```



[ ]:
```
#load model
from tensorflow.keras.models import load_model

pathModel = '/content/drive/My Drive/ColabNotebooks/Project/
 ↪predictive_analysis_of_Total_number_in_wage_employment_model.h5'

model = load_model(pathModel)
```

7.1.12 Making a Prediction

[ ]:
```
# make a prediction
yhat = model.predict(test_X)

#reshaping test_X
test_x = test_X.reshape((test_X.shape[0], test_X.shape[2]))

# invert scaling for forecast
inv_yhat = concatenate((yhat, test_x[:, 1:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:,0]

# invert scaling for actual
test_y = test_y.reshape((len(test_y), 1))
inv_y = concatenate((test_y, test_x[:, 1:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
```

```
inv_y = inv_y[:,0]
```

7.1.13 Displaying Predicted Values: Total Number of People in Wage Employment

[ ]:
```
# Type casting predictions to int
inv_yhat = inv_yhat.astype(int)

# Displaying predicted values
print('Predicted values are: ')
print(inv_yhat)
```

```
Predicted values are:
[ 11792 312891  18415  23090 147698 237846  89797  75574  86442  74913
   8283  68559   8950 232003 482441 128458   9634  32995 121926  14529]
```

8 Model Evaluation of Model 2: Predictive Analysis of Total Employment

8.0.1 Evaluating model using RMSE

[ ]:
```
# calculating RMSE
rmse = sqrt(mean_squared_error(inv_y, inv_yhat))
print('Root Mean Square Error: %.3f' % rmse)

#Normalizing RMSE using range
nrmse = rmse/ (inv_y.max() - inv_yhat.min())
print('Normalized Root Mean Square Error: %.3f' %nrmse)

#Accuracy of model
accuracy = 1-nrmse
print("Accuracy of model is: %.3f " % accuracy)
```

```
Root Mean Square Error: 38460.426
Normalized Root Mean Square Error: 0.068
Accuracy of model is: 0.932
```

Normalizing the RMSE in the range (0,1) yields an NRMSE of 0.059, which is considerably low, hence showing the model perfomed very well. With subtracting the NRMSE from 1, I can loosely say the model has an accuracy of about 94.1 %

8.0.2 Plotting Actual vs Forecast Values

[ ]:
```
import matplotlib.pyplot as plt
plt.plot(inv_y, color = 'black', label = 'Actual')
plt.plot(inv_yhat, color = 'Green', label = 'Predicted')
plt.title('Wage Employment')
```

```
plt.xlabel('Industry Columns')
plt.ylabel('Total_number_in_wage_employment')
plt.legend()
plt.show()
```

The plots of the actual vs the forecast values are closely knit, depicting that the model was able to capture the trend of the total number of people in wage emplyment according to their Industry/Sector of employment, based on the input at the prior time step

[ ]:

8.1 -------------------------------- End of Part 5 and 6 (Modelling and Model Evaluation -------------

[ ]:

[ ]:

9 Part 7: Model Deployment

[ ]:
```
!pip install flask
```

```
Requirement already satisfied: flask in /usr/local/lib/python3.6/dist-packages
(1.1.2)
Requirement already satisfied: Jinja2>=2.10.1 in /usr/local/lib/python3.6/dist-
packages (from flask) (2.11.2)
Requirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python3.6/dist-
packages (from flask) (1.0.1)
Requirement already satisfied: itsdangerous>=0.24 in /usr/local/lib/python3.6
/dist-packages (from flask) (1.1.0)
Requirement already satisfied: click>=5.1 in /usr/local/lib/python3.6/dist-
packages (from flask) (7.1.2)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6
/dist-packages (from Jinja2>=2.10.1->flask) (1.1.1)
```

[ ]:
```
!pip install flask-wtf
```

```
Collecting flask-wtf
  Downloading https://files.pythonhosted.org/packages/36/a9/8c01171066bd7a524ee0
05d81bb4a8aa446ab178043a1ad6cb5dc8f0bd83/Flask_WTF-0.14.3-py2.py3-none-any.whl
Requirement already satisfied: Flask in /usr/local/lib/python3.6/dist-packages
(from flask-wtf) (1.1.2)
Requirement already satisfied: itsdangerous in /usr/local/lib/python3.6/dist-
packages (from flask-wtf) (1.1.0)
Collecting WTForms
  Downloading https://files.pythonhosted.org/packages/e0/31/614fc7dc7d7600
5b0acb8c0c8920d962b83d7422b4ba912886dfb63f86ff/WTForms-2.3.3-py2.py3-none-
any.whl (169kB)
     || 174kB 7.7MB/s
Requirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python3.6
/dist-packages (from Flask->flask-wtf) (1.0.1)
Requirement already satisfied: click>=5.1 in /usr/local/lib/python3.6/dist-
packages (from Flask->flask-wtf) (7.1.2)
Requirement already satisfied: Jinja2>=2.10.1 in /usr/local/lib/python3.6/dist-
packages (from Flask->flask-wtf) (2.11.2)
Requirement already satisfied: MarkupSafe in /usr/local/lib/python3.6/dist-
packages (from WTForms->flask-wtf) (1.1.1)
Installing collected packages: WTForms, flask-wtf
Successfully installed WTForms-2.3.3 flask-wtf-0.14.3
```

[ ]:
```
!pip install gunicorn
```

```
Collecting gunicorn
  Downloading https://files.pythonhosted.org/packages/69/ca/926f7cd3a2014b
16870086b2d0fdc84a9e49473c68a8dff8b57f7c156f43/gunicorn-20.0.4-py2.py3-none-
any.whl (77kB)
     || 81kB 5.3MB/s
Requirement already satisfied: setuptools>=3.0 in /usr/local/lib/python3.6
/dist-packages (from gunicorn) (51.0.0)
```

```
Installing collected packages: gunicorn
Successfully installed gunicorn-20.0.4
```

[ ]:
```
!pip freeze > requirements.txt
```

[ ]:
```

```

10 PDF Generation

[1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[2]:
```
cd  /content/drive/My Drive/ColabNotebooks/Project/
```

```
/content/drive/My Drive/ColabNotebooks/Project
```

[ ]:
```
!apt-get install texlive-xetex texlive-fonts-recommended␣
 ↪texlive-generic-recommended
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive-base
  texlive-binaries texlive-latex-base texlive-latex-extra
  texlive-latex-recommended texlive-pictures texlive-plain-generic tipa
Suggested packages:
  fonts-noto apache2 | lighttpd | httpd poppler-utils ghostscript
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic
  | fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri
  ruby-dev bundler debhelper gv | postscript-viewer perl-tk xpdf-reader
  | pdf-viewer texlive-fonts-recommended-doc texlive-latex-base-doc
  python-pygments icc-profiles libfile-which-perl
  libspreadsheet-parseexcel-perl texlive-latex-extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk
  | libtcltk-ruby texlive-pictures-doc vprerex
The following NEW packages will be installed:
```

```
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive-base
  texlive-binaries texlive-fonts-recommended texlive-generic-recommended
  texlive-latex-base texlive-latex-extra texlive-latex-recommended
  texlive-pictures texlive-plain-generic texlive-xetex tipa
0 upgraded, 47 newly installed, 0 to remove and 15 not upgraded.
Need to get 146 MB of archives.
After this operation, 460 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-droid-fallback
all 1:6.0.1r16-1.1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lato all 2.0-2
[2,698 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 poppler-data all
0.4.8-2 [1,479 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 tex-common all 6.09
[33.0 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lmodern all
2.004.5-3 [4,551 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-noto-mono all
20171026-2 [75.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 fonts-texgyre all
20160520-1 [8,761 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 javascript-common all
11 [6,066 B]
Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsfilters1
amd64 1.20.2-0ubuntu3.1 [108 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsimage2
amd64 2.2.7-1ubuntu2.8 [18.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/main amd64 libijs-0.35 amd64
0.35-13 [15.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjbig2dec0 amd64
0.13-6 [55.9 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9-common
all 9.26~dfsg+0-0ubuntu0.18.04.14 [5,092 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9 amd64
9.26~dfsg+0-0ubuntu0.18.04.14 [2,265 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjs-jquery all
3.2.1-1 [152 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libkpathsea6
amd64 2017.20170613.44572-8ubuntu0.1 [54.9 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpotrace0 amd64
1.14-2 [17.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libptexenc1
```

```
amd64 2017.20170613.44572-8ubuntu0.1 [34.5 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic/main amd64 rubygems-integration
all 1.11 [4,994 B]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 ruby2.5 amd64
2.5.1-1ubuntu1.7 [48.6 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby amd64 1:2.5.1
[5,712 B]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 rake all
12.3.1-1ubuntu0.1 [44.9 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-did-you-mean all
1.2.0-2 [9,700 B]
Get:24 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-minitest all
5.10.3-1 [38.6 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:26 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-power-assert all
0.3.0-1 [7,952 B]
Get:27 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-test-unit all
3.2.5-1 [61.1 kB]
Get:28 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libruby2.5
amd64 2.5.1-1ubuntu1.7 [3,068 kB]
Get:29 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsynctex1
amd64 2017.20170613.44572-8ubuntu0.1 [41.4 kB]
Get:30 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexlua52
amd64 2017.20170613.44572-8ubuntu0.1 [91.2 kB]
Get:31 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexluajit2
amd64 2017.20170613.44572-8ubuntu0.1 [230 kB]
Get:32 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libzzip-0-13
amd64 0.13.62-3.1ubuntu0.18.04.1 [26.0 kB]
Get:33 http://archive.ubuntu.com/ubuntu bionic/main amd64 lmodern all 2.004.5-3
[9,631 kB]
Get:34 http://archive.ubuntu.com/ubuntu bionic/main amd64 preview-latex-style
all 11.91-1ubuntu1 [185 kB]
Get:35 http://archive.ubuntu.com/ubuntu bionic/main amd64 t1utils amd64 1.41-2
[56.0 kB]
Get:36 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tex-gyre all
20160520-1 [4,998 kB]
Get:37 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 texlive-
binaries amd64 2017.20170613.44572-8ubuntu0.1 [8,179 kB]
Get:38 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-base all
2017.20180305-1 [18.7 MB]
Get:39 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-fonts-
recommended all 2017.20180305-1 [5,262 kB]
Get:40 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-plain-
generic all 2017.20180305-2 [23.6 MB]
Get:41 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-generic-
recommended all 2017.20180305-1 [15.9 kB]
Get:42 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-base all
```

```
2017.20180305-1 [951 kB]
Get:43 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-
recommended all 2017.20180305-1 [14.9 MB]
Get:44 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-pictures
all 2017.20180305-1 [4,026 kB]
Get:45 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-latex-
extra all 2017.20180305-2 [10.6 MB]
Get:46 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tipa all 2:1.3-20
[2,978 kB]
Get:47 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-xetex all
2017.20180305-1 [10.7 MB]
Fetched 146 MB in 2s (65.7 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages ...
Selecting previously unselected package fonts-droid-fallback.
(Reading database ... 146442 files and directories currently installed.)
Preparing to unpack .../00-fonts-droid-fallback_1%3a6.0.1r16-1.1_all.deb ...
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1) ...
Selecting previously unselected package fonts-lato.
Preparing to unpack .../01-fonts-lato_2.0-2_all.deb ...
Unpacking fonts-lato (2.0-2) ...
Selecting previously unselected package poppler-data.
Preparing to unpack .../02-poppler-data_0.4.8-2_all.deb ...
Unpacking poppler-data (0.4.8-2) ...
Selecting previously unselected package tex-common.
Preparing to unpack .../03-tex-common_6.09_all.deb ...
Unpacking tex-common (6.09) ...
Selecting previously unselected package fonts-lmodern.
Preparing to unpack .../04-fonts-lmodern_2.004.5-3_all.deb ...
Unpacking fonts-lmodern (2.004.5-3) ...
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack .../05-fonts-noto-mono_20171026-2_all.deb ...
Unpacking fonts-noto-mono (20171026-2) ...
Selecting previously unselected package fonts-texgyre.
Preparing to unpack .../06-fonts-texgyre_20160520-1_all.deb ...
Unpacking fonts-texgyre (20160520-1) ...
Selecting previously unselected package javascript-common.
Preparing to unpack .../07-javascript-common_11_all.deb ...
Unpacking javascript-common (11) ...
Selecting previously unselected package libcupsfilters1:amd64.
Preparing to unpack .../08-libcupsfilters1_1.20.2-0ubuntu3.1_amd64.deb ...
Unpacking libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) ...
Selecting previously unselected package libcupsimage2:amd64.
Preparing to unpack .../09-libcupsimage2_2.2.7-1ubuntu2.8_amd64.deb ...
Unpacking libcupsimage2:amd64 (2.2.7-1ubuntu2.8) ...
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack .../10-libijs-0.35_0.35-13_amd64.deb ...
Unpacking libijs-0.35:amd64 (0.35-13) ...
```

```
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack .../11-libjbig2dec0_0.13-6_amd64.deb ...
Unpacking libjbig2dec0:amd64 (0.13-6) ...
Selecting previously unselected package libgs9-common.
Preparing to unpack .../12-libgs9-common_9.26~dfsg+0-0ubuntu0.18.04.14_all.deb
...
Unpacking libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.14) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../13-libgs9_9.26~dfsg+0-0ubuntu0.18.04.14_amd64.deb ...
Unpacking libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.14) ...
Selecting previously unselected package libjs-jquery.
Preparing to unpack .../14-libjs-jquery_3.2.1-1_all.deb ...
Unpacking libjs-jquery (3.2.1-1) ...
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack .../15-libkpathsea6_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libpotrace0.
Preparing to unpack .../16-libpotrace0_1.14-2_amd64.deb ...
Unpacking libpotrace0 (1.14-2) ...
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack .../17-libptexenc1_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../18-rubygems-integration_1.11_all.deb ...
Unpacking rubygems-integration (1.11) ...
Selecting previously unselected package ruby2.5.
Preparing to unpack .../19-ruby2.5_2.5.1-1ubuntu1.7_amd64.deb ...
Unpacking ruby2.5 (2.5.1-1ubuntu1.7) ...
Selecting previously unselected package ruby.
Preparing to unpack .../20-ruby_1%3a2.5.1_amd64.deb ...
Unpacking ruby (1:2.5.1) ...
Selecting previously unselected package rake.
Preparing to unpack .../21-rake_12.3.1-1ubuntu0.1_all.deb ...
Unpacking rake (12.3.1-1ubuntu0.1) ...
Selecting previously unselected package ruby-did-you-mean.
Preparing to unpack .../22-ruby-did-you-mean_1.2.0-2_all.deb ...
Unpacking ruby-did-you-mean (1.2.0-2) ...
Selecting previously unselected package ruby-minitest.
Preparing to unpack .../23-ruby-minitest_5.10.3-1_all.deb ...
Unpacking ruby-minitest (5.10.3-1) ...
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack .../24-ruby-net-telnet_0.1.1-2_all.deb ...
Unpacking ruby-net-telnet (0.1.1-2) ...
Selecting previously unselected package ruby-power-assert.
Preparing to unpack .../25-ruby-power-assert_0.3.0-1_all.deb ...
Unpacking ruby-power-assert (0.3.0-1) ...
```

```
Selecting previously unselected package ruby-test-unit.
Preparing to unpack .../26-ruby-test-unit_3.2.5-1_all.deb ...
Unpacking ruby-test-unit (3.2.5-1) ...
Selecting previously unselected package libruby2.5:amd64.
Preparing to unpack .../27-libruby2.5_2.5.1-1ubuntu1.7_amd64.deb ...
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.7) ...
Selecting previously unselected package libsynctex1:amd64.
Preparing to unpack .../28-libsynctex1_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libtexlua52:amd64.
Preparing to unpack .../29-libtexlua52_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
.../30-libtexluajit2_2017.20170613.44572-8ubuntu0.1_amd64.deb ...
Unpacking libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack .../31-libzzip-0-13_0.13.62-3.1ubuntu0.18.04.1_amd64.deb ...
Unpacking libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) ...
Selecting previously unselected package lmodern.
Preparing to unpack .../32-lmodern_2.004.5-3_all.deb ...
Unpacking lmodern (2.004.5-3) ...
Selecting previously unselected package preview-latex-style.
Preparing to unpack .../33-preview-latex-style_11.91-1ubuntu1_all.deb ...
Unpacking preview-latex-style (11.91-1ubuntu1) ...
Selecting previously unselected package t1utils.
Preparing to unpack .../34-t1utils_1.41-2_amd64.deb ...
Unpacking t1utils (1.41-2) ...
Selecting previously unselected package tex-gyre.
Preparing to unpack .../35-tex-gyre_20160520-1_all.deb ...
Unpacking tex-gyre (20160520-1) ...
Selecting previously unselected package texlive-binaries.
Preparing to unpack .../36-texlive-
binaries_2017.20170613.44572-8ubuntu0.1_amd64.deb ...
Unpacking texlive-binaries (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package texlive-base.
Preparing to unpack .../37-texlive-base_2017.20180305-1_all.deb ...
Unpacking texlive-base (2017.20180305-1) ...
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack .../38-texlive-fonts-recommended_2017.20180305-1_all.deb ...
Unpacking texlive-fonts-recommended (2017.20180305-1) ...
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack .../39-texlive-plain-generic_2017.20180305-2_all.deb ...
Unpacking texlive-plain-generic (2017.20180305-2) ...
Selecting previously unselected package texlive-generic-recommended.
Preparing to unpack .../40-texlive-generic-recommended_2017.20180305-1_all.deb
```

```
...
Unpacking texlive-generic-recommended (2017.20180305-1) ...
Selecting previously unselected package texlive-latex-base.
Preparing to unpack .../41-texlive-latex-base_2017.20180305-1_all.deb ...
Unpacking texlive-latex-base (2017.20180305-1) ...
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack .../42-texlive-latex-recommended_2017.20180305-1_all.deb ...
Unpacking texlive-latex-recommended (2017.20180305-1) ...
Selecting previously unselected package texlive-pictures.
Preparing to unpack .../43-texlive-pictures_2017.20180305-1_all.deb ...
Unpacking texlive-pictures (2017.20180305-1) ...
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack .../44-texlive-latex-extra_2017.20180305-2_all.deb ...
Unpacking texlive-latex-extra (2017.20180305-2) ...
Selecting previously unselected package tipa.
Preparing to unpack .../45-tipa_2%3a1.3-20_all.deb ...
Unpacking tipa (2:1.3-20) ...
Selecting previously unselected package texlive-xetex.
Preparing to unpack .../46-texlive-xetex_2017.20180305-1_all.deb ...
Unpacking texlive-xetex (2017.20180305-1) ...
Setting up libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.14) ...
Setting up libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up libjs-jquery (3.2.1-1) ...
Setting up libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up fonts-droid-fallback (1:6.0.1r16-1.1) ...
Setting up libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up tex-common (6.09) ...
update-language: texlive-base not installed and configured, doing nothing!
Setting up poppler-data (0.4.8-2) ...
Setting up tex-gyre (20160520-1) ...
Setting up preview-latex-style (11.91-1ubuntu1) ...
Setting up fonts-texgyre (20160520-1) ...
Setting up fonts-noto-mono (20171026-2) ...
Setting up fonts-lato (2.0-2) ...
Setting up libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) ...
Setting up libcupsimage2:amd64 (2.2.7-1ubuntu2.8) ...
Setting up libjbig2dec0:amd64 (0.13-6) ...
Setting up ruby-did-you-mean (1.2.0-2) ...
Setting up t1utils (1.41-2) ...
Setting up ruby-net-telnet (0.1.1-2) ...
Setting up libijs-0.35:amd64 (0.35-13) ...
Setting up rubygems-integration (1.11) ...
Setting up libpotrace0 (1.14-2) ...
Setting up javascript-common (11) ...
Setting up ruby-minitest (5.10.3-1) ...
Setting up libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) ...
Setting up libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.14) ...
```

```
Setting up libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Setting up fonts-lmodern (2.004.5-3) ...
Setting up ruby-power-assert (0.3.0-1) ...
Setting up texlive-binaries (2017.20170613.44572-8ubuntu0.1) ...
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up texlive-base (2017.20180305-1) ...
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST...
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN...
mktexlsr: Updating /var/lib/texmf/ls-R...
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4: /var/lib/texmf/dvips/config
/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4: /var/lib/texmf/dvipdfmx
/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4:
/var/lib/texmf/tex/generic/config/pdftexconfig.tex
Setting up texlive-fonts-recommended (2017.20180305-1) ...
Setting up texlive-plain-generic (2017.20180305-2) ...
Setting up texlive-generic-recommended (2017.20180305-1) ...
Setting up texlive-latex-base (2017.20180305-1) ...
Setting up lmodern (2.004.5-3) ...
Setting up texlive-latex-recommended (2017.20180305-1) ...
Setting up texlive-pictures (2017.20180305-1) ...
Setting up tipa (2:1.3-20) ...
Regenerating '/var/lib/texmf/fmtutil.cnf-DEBIAN'... done.
Regenerating '/var/lib/texmf/fmtutil.cnf-TEXLIVEDIST'... done.
update-fmtutil has updated the following file(s):
        /var/lib/texmf/fmtutil.cnf-DEBIAN
        /var/lib/texmf/fmtutil.cnf-TEXLIVEDIST
If you want to activate the changes in the above file(s),
you should run fmtutil-sys or fmtutil.
Setting up texlive-latex-extra (2017.20180305-2) ...
Setting up texlive-xetex (2017.20180305-1) ...
Setting up ruby2.5 (2.5.1-1ubuntu1.7) ...
Setting up ruby (1:2.5.1) ...
Setting up ruby-test-unit (3.2.5-1) ...
Setting up rake (12.3.1-1ubuntu0.1) ...
Setting up libruby2.5:amd64 (2.5.1-1ubuntu1.7) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.3) ...
/sbin/ldconfig.real: /usr/local/lib/python3.6/dist-
packages/ideep4py/lib/libmkldnn.so.0 is not a symbolic link

Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

```
Processing triggers for fontconfig (2.12.6-0ubuntu2) ...
Processing triggers for tex-common (6.09) ...
Running updmap-sys. This may take some time... done.
Running mktexlsr /var/lib/texmf ... done.
Building format(s) --all.
        This may take some time... done.
```

[3]:
```
!jupyter nbconvert --to pdf␣
↪Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth.ipynb
```

```
[NbConvertApp] Converting notebook
Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth.ipynb to
pdf
[NbConvertApp] Support files will be in
Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files/
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
[NbConvertApp] Making directory
./Predictive_Analysis_of_Productive_Employment_based_on_Economic_Growth_files
```

```
[NbConvertApp] Writing 219195 bytes to ./notebook.tex
[NbConvertApp] Building PDF
Traceback (most recent call last):
  File "/usr/local/bin/jupyter-nbconvert", line 8, in <module>
    sys.exit(main())
  File "/usr/local/lib/python2.7/dist-packages/jupyter_core/application.py",
line 267, in launch_instance
    return super(JupyterApp, cls).launch_instance(argv=argv, **kwargs)
  File "/usr/local/lib/python2.7/dist-packages/traitlets/config/application.py",
line 658, in launch_instance
    app.start()
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/nbconvertapp.py", line
338, in start
    self.convert_notebooks()
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/nbconvertapp.py", line
508, in convert_notebooks
    self.convert_single_notebook(notebook_filename)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/nbconvertapp.py", line
479, in convert_single_notebook
    output, resources = self.export_single_notebook(notebook_filename,
resources, input_buffer=input_buffer)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/nbconvertapp.py", line
408, in export_single_notebook
    output, resources = self.exporter.from_filename(notebook_filename,
resources=resources)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/exporter.py",
line 179, in from_filename
    return self.from_file(f, resources=resources, **kw)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/exporter.py",
line 197, in from_file
    return self.from_notebook_node(nbformat.read(file_stream, as_version=4),
resources=resources, **kw)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/pdf.py", line
178, in from_notebook_node
    rc = self.run_latex(tex_file)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/pdf.py", line
149, in run_latex
    self.latex_count, log_error)
  File "/usr/local/lib/python2.7/dist-packages/nbconvert/exporters/pdf.py", line
111, in run_command
    "at {link}.".format(formatter=command_list[0], link=link))
OSError: xelatex not found on PATH, if you have not installed xelatex you may
need to do so. Find further instructions at
https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex.
```