In [1]:

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

In [2]:

```
data=pd.read_csv(r"C:\Users\user\Downloads\sector_dataset_revised.csv")
data
```

Out[2]:

	country	country_code	sector	sector_number	year	Nominal_value_adde
0	Australia	AUS	Total	0	1975	
1	Australia	AUS	Agriculture	1	1975	
2	Australia	AUS	Mining	2	1975	
3	Australia	AUS	Manufacturing	3	1975	
4	Australia	AUS	Utilities	4	1975	
...	...	...	...	...	...	
35149	Uganda	UGA	Utilities	4	2018	4
35150	Uganda	UGA	Construction	5	2018	8
35151	Uganda	UGA	Whole sale, Accommodation and food service act...	6	2018	18
35152	Uganda	UGA	Transportation,information and communication, ...	7	2018	18
35153	Uganda	UGA	Government services, Community, social and per...	8	2018	16

35154 rows × 11 columns

In [3]:
```
da=data.head(100)
da
```

Out[3]:

_number	year	Nominal_value_added_LCU	Nominal_value_added_USD	Real_value_added_LCU	Em
0	1975	76723	100440	427474	
1	1975	4291	5617	18379	
2	1975	3801	4976	16013	
3	1975	15253	19968	65863	
4	1975	2479	3246	15131	
...	...	...	...	...	
5	1985	16856	11772	42433	
6	1985	28813	20122	69052	
7	1985	63330	44228	177417	
8	1985	52131	36407	149734	
0	1986	263135	175896	602233	

In [4]:
```
data.info()
```

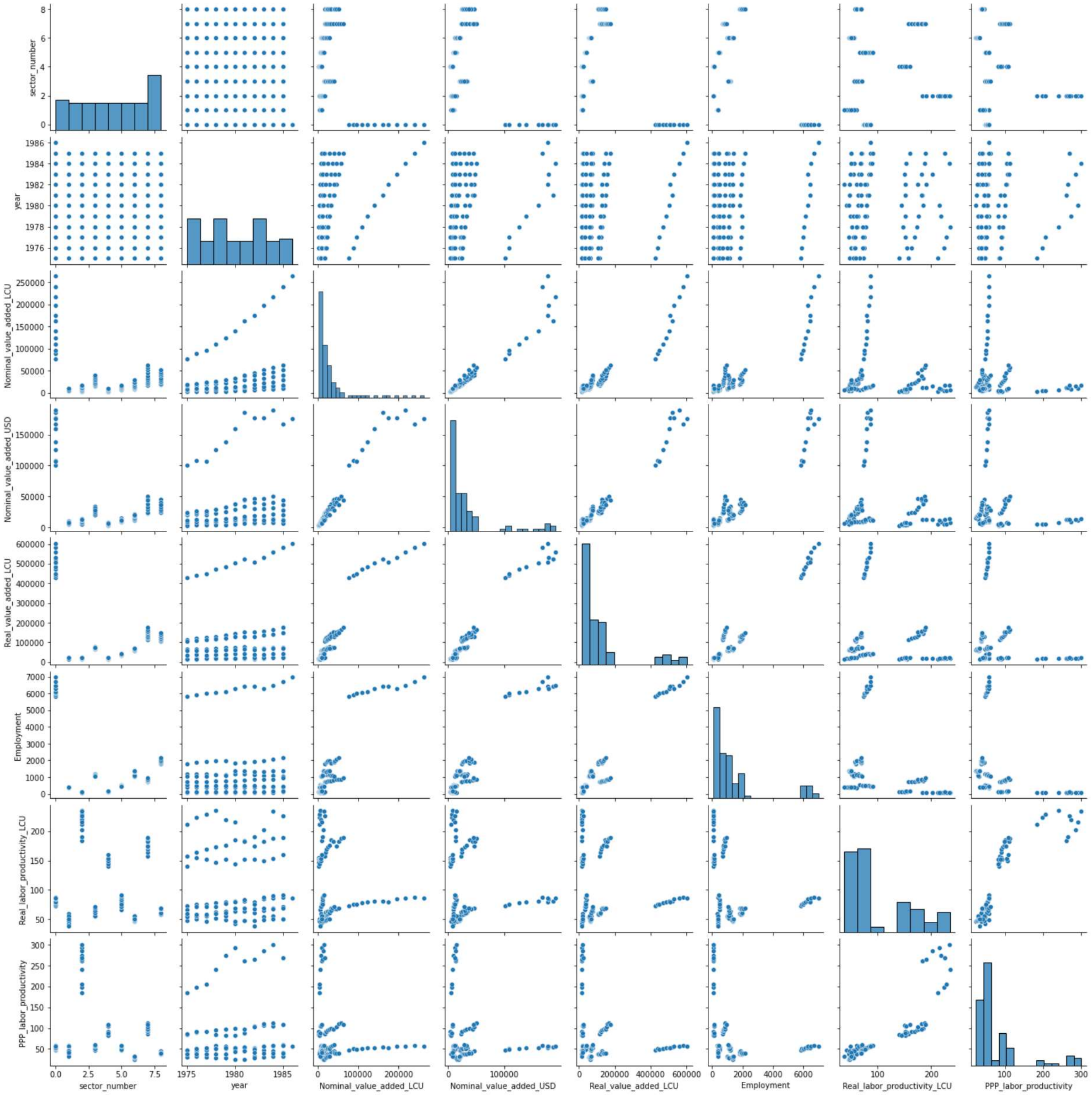
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35154 entries, 0 to 35153
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype 
---  ------                      --------------  ----- 
 0   country                     35154 non-null  object
 1   country_code                35154 non-null  object
 2   sector                      35154 non-null  object
 3   sector_number               35154 non-null  int64 
 4   year                        35154 non-null  int64 
 5   Nominal_value_added_LCU     35154 non-null  int64 
 6   Nominal_value_added_USD     35154 non-null  int64 
 7   Real_value_added_LCU        35154 non-null  int64 
 8   Employment                  35154 non-null  int64 
 9   Real_labor_productivity_LCU  35154 non-null  int64 
 10  PPP_labor_productivity      35154 non-null  int64 
dtypes: int64(8), object(3)
memory usage: 3.0+ MB
```

In [5]:
```
da.columns
```

Out[5]:
```
Index(['country', 'country_code', 'sector', 'sector_number', 'year',
       'Nominal_value_added_LCU', 'Nominal_value_added_USD',
       'Real_value_added_LCU', 'Employment', 'Real_labor_productivity_LCU',
       'PPP_labor_productivity'],
      dtype='object')
```

In [6]:
```
sns.pairplot(da)
```

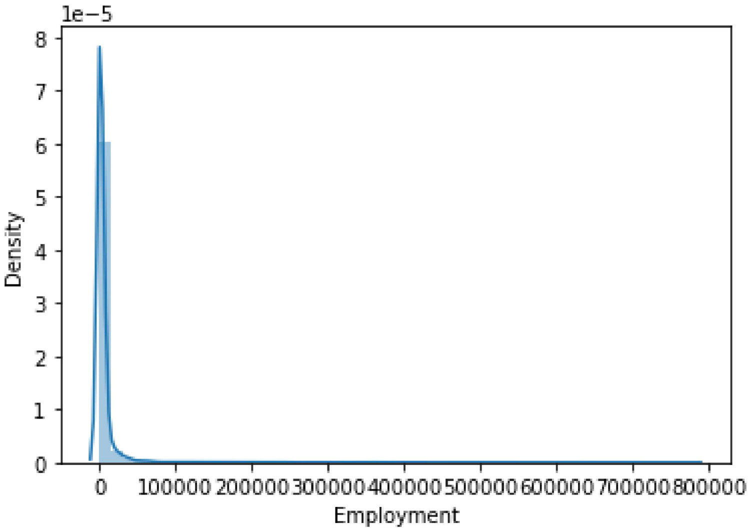
Out[6]:
```
<seaborn.axisgrid.PairGrid at 0x167017f2ca0>
```



In [9]:
```
sns.distplot(data["Employment"])
```

```
C:\Users\USER\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futu
reWarning: `distplot` is a deprecated function and will be removed in a futur
e version. Please adapt your code to use either `displot` (a figure-level fun
ction with similar flexibility) or `histplot` (an axes-level function for his
tograms).
  warnings.warn(msg, FutureWarning)
```

Out[9]:
```
<AxesSubplot:xlabel='Employment', ylabel='Density'>
```



Linear Regression

In [10]:
```
df=da[['sector_number', 'year',
       'Nominal_value_added_LCU', 'Nominal_value_added_USD',
       'Real_value_added_LCU', 'Employment', 'Real_labor_productivity_LCU',
       'PPP_labor_productivity']]
```

In [11]:
```
x=df[['sector_number', 'year',
       'Nominal_value_added_LCU', 'Nominal_value_added_USD',
       'Real_value_added_LCU', 'Employment', 'Real_labor_productivity_LCU']]
y=df[['PPP_labor_productivity']]
```

In [13]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [16]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

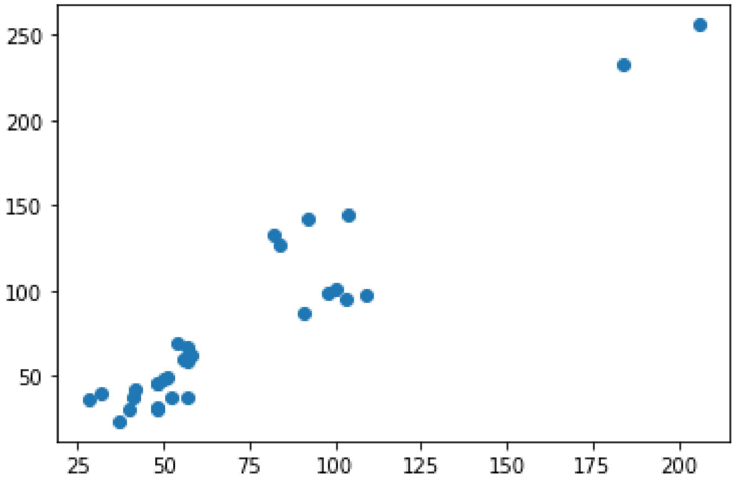
Out[16]:
```
LinearRegression()
```

```
In [17]: print(lr.intercept_)
```

```
[-838.75566248]
```

```
In [18]: prediction=lr.predict(x_test)
         plt.scatter(y_test,prediction)
```

```
Out[18]: <matplotlib.collections.PathCollection at 0x1670a7bbe80>
```



```
In [19]: print(lr.score(x_test,y_test))
```

```
0.6817567228468826
```

Ridge Regression

```
In [21]: from sklearn.linear_model import Ridge
```

```
In [22]: rr=Ridge(alpha=0)
         rr.fit(x_train,y_train)
```

```
Out[22]: Ridge(alpha=0)
```

```
In [23]: rr.score(x_test,y_test)
```

```
Out[23]: 0.6817567228468757
```

Lasso Regression

```
In [24]: from sklearn.linear_model import Lasso
```

In [25]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[25]:
```
Lasso(alpha=10)
```

In [26]:
```
la.score(x_test,y_test)
```

Out[26]:
```
0.6646662695472622
```

Elastic regression

In [27]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[27]:
```
ElasticNet()
```

In [28]:
```
print(en.coef_)
```

```
[-4.20980301e+00  2.25776857e-01  4.12207102e-04  5.81578865e-04
 -1.18897592e-03  6.93498212e-02  1.38020131e+00]
```

In [29]:
```
print(en.score(x_test,y_test))
```

```
0.6776894621477989
```

Logistic Regression

In [55]:
```
from sklearn.linear_model import LogisticRegression
```

In [56]:
```
df=da[['sector_number', 'year',
       'Nominal_value_added_LCU', 'Nominal_value_added_USD',
       'Real_value_added_LCU', 'Employment', 'Real_labor_productivity_LCU',
       'PPP_labor_productivity']]
```

In [57]:
```
feature_matrix=df.iloc[:,0:11]
target_vector=df.iloc[:,-1]
```

In [58]:
```
feature_matrix.shape
```

Out[58]:
```
(100, 8)
```

In [59]:
```
target_vector.shape
```

Out[59]:
```
(100,)
```

In [60]:
```
from sklearn.preprocessing import StandardScaler
```

In [61]:
```
fs=StandardScaler().fit_transform(feature_matrix)
```

In [65]:
```
logr=LogisticRegression()
logr.fit(fs,target_vector)
```

Out[65]:
```
LogisticRegression()
```

In [68]:
```
observation=[[1,2,3,4,5,6,7,8]]
```

In [69]:
```
predicton=logr.predict(observation)
print(observation)
```

```
[[1, 2, 3, 4, 5, 6, 7, 8]]
```

In [71]:
```
logr.classes_
```

Out[71]:
```
array([ 24,  25,  28,  29,  30,  31,  32,  33,  37,  38,  39,  40,  41,
        42,  43,  45,  46,  47,  48,  49,  50,  51,  52,  53,  54,  55,
        56,  57,  58,  61,  82,  83,  84,  85,  86,  88,  90,  91,  92,
        95,  98, 100, 103, 104, 106, 108, 109, 111, 112, 184, 198, 206,
       241, 262, 265, 269, 274, 285, 293, 301], dtype=int64)
```

In [72]:
```
logr.score(fs,target_vector)
```

Out[72]:
```
0.31
```

Random Forest

In [74]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import plot_tree
```

In [76]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)
```

In [77]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_2068\2068597404.py:2: DataConversi
onWarning: A column-vector y was passed when a 1d array was expected. Please
change the shape of y to (n_samples,), for example using ravel().
  rfc.fit(x_train,y_train)
```

Out[77]:
```
RandomForestClassifier()
```

In [82]:
```
parameters={'max_depth':[1,2,3,4,5],
           'min_samples_leaf':[5,6,7,8,9,10],
           'n_estimators':[10,20,30,40,50]}
```

In [83]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring='accu
grid_search.fit(x_train,y_train)
```

```
C:\Users\USER\anaconda3\lib\site-packages\sklearn\model_selection\_split.p
y:676: UserWarning: The least populated class in y has only 1 members, whic
h is less than n_splits=2.
  warnings.warn(
C:\Users\USER\anaconda3\lib\site-packages\sklearn\model_selection\_validati
on.py:680: DataConversionWarning: A column-vector y was passed when a 1d ar
ray was expected. Please change the shape of y to (n_samples,), for example
using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\USER\anaconda3\lib\site-packages\sklearn\model_selection\_validati
on.py:680: DataConversionWarning: A column-vector y was passed when a 1d ar
ray was expected. Please change the shape of y to (n_samples,), for example
using ravel().
  estimator.fit(X_train, y_train, **fit_params)
C:\Users\USER\anaconda3\lib\site-packages\sklearn\model_selection\_validati
on.py:680: DataConversionWarning: A column-vector y was passed when a 1d ar
ray was expected. Please change the shape of y to (n_samples,), for example
using ravel().
  estimator.fit(X_train, y_train, **fit_params)
```

In [84]:
```
grid_search.best_score_
```

Out[84]:
```
0.12857142857142856
```

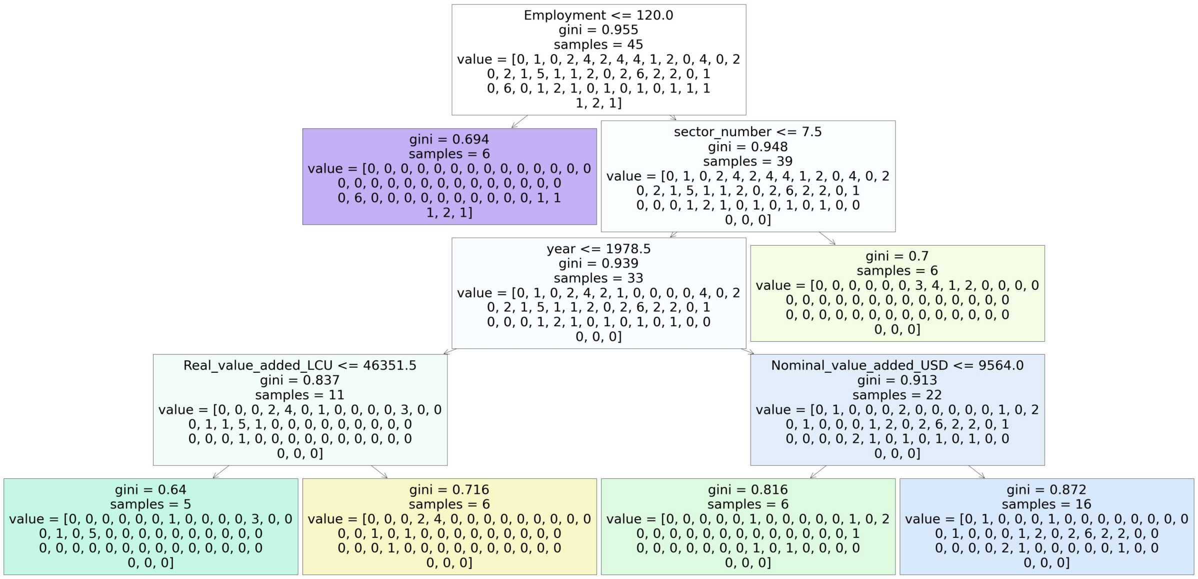
In [85]:
```
rfc_best=grid_search.best_estimator_
```

In [86]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,filled=True)
```

Out[86]:

```
[Text(0.5, 0.9, 'Employment <= 120.0\ngini = 0.955\nsamples = 45\nvalue = [0,
1, 0, 2, 4, 2, 4, 4, 1, 2, 0, 4, 0, 2\n0, 2, 1, 5, 1, 1, 2, 0, 2, 6, 2, 2, 0,
1\n0, 6, 0, 1, 2, 1, 0, 1, 0, 1, 0, 1, 1, 1\n1, 2, 1]'),
 Text(0.375, 0.7, 'gini = 0.694\nsamples = 6\nvalue = [0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 6, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 1\n1, 2, 1]'),
 Text(0.625, 0.7, 'sector_number <= 7.5\ngini = 0.948\nsamples = 39\nvalue =
[0, 1, 0, 2, 4, 2, 4, 4, 1, 2, 0, 4, 0, 2\n0, 2, 1, 5, 1, 1, 2, 0, 2, 6, 2,
2, 0, 1\n0, 0, 0, 1, 2, 1, 0, 1, 0, 1, 0, 1, 0, 0\n0, 0]'),
 Text(0.5, 0.5, 'year <= 1978.5\ngini = 0.939\nsamples = 33\nvalue = [0, 1,
0, 2, 4, 2, 1, 0, 0, 0, 0, 4, 0, 2\n0, 2, 1, 5, 1, 1, 2, 0, 2, 6, 2, 2, 0, 1
\n0, 0, 0, 1, 2, 1, 0, 1, 0, 1, 0, 1, 0, 0\n0, 0, 0]'),
 Text(0.25, 0.3, 'Real_value_added_LCU <= 46351.5\ngini = 0.837\nsamples = 11
\nvalue = [0, 0, 0, 2, 4, 0, 1, 0, 0, 0, 0, 3, 0, 0\n0, 1, 1, 5, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0\n0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.125, 0.1, 'gini = 0.64\nsamples = 5\nvalue = [0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 3, 0, 0\n0, 1, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.375, 0.1, 'gini = 0.716\nsamples = 6\nvalue = [0, 0, 0, 2, 4, 0, 0,
0, 0, 0, 0, 0, 0, 0\n0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.75, 0.3, 'Nominal_value_added_USD <= 9564.0\ngini = 0.913\nsamples =
22\nvalue = [0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 2\n0, 1, 0, 0, 0, 1, 2,
0, 2, 6, 2, 2, 0, 1\n0, 0, 0, 0, 2, 1, 0, 1, 0, 1, 0, 1, 0, 0\n0, 0, 0]'),
 Text(0.625, 0.1, 'gini = 0.816\nsamples = 6\nvalue = [0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 1, 0, 2\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1\n0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.875, 0.1, 'gini = 0.872\nsamples = 16\nvalue = [0, 1, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0\n0, 1, 0, 0, 0, 1, 2, 0, 2, 6, 2, 2, 0, 0\n0, 0, 0, 0, 2,
1, 0, 0, 0, 0, 0, 1, 0, 0\n0, 0, 0]'),
 Text(0.75, 0.5, 'gini = 0.7\nsamples = 6\nvalue = [0, 0, 0, 0, 0, 0, 3, 4,
1, 2, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0]')]
```



In [ ]: