BSTAT 5325

PROJECT

PREDICTING THE PROSPERITY OF COUNTRIES

MEKAPATI DARSHANIK

UTA ID: 1001904282

AUGUST- 2, 2021

CONTENTS

MOTIVATION:

A country's prosperity is the measure that tells the country's Economic and lifestyle conditions and its overall growth. There were many factors(variables) that indicate the country's prosperity. Some of them are as follows:

1. GDP Per capita
2. Inflation Rate
3. Political Participation
4. Political Culture
5. Population
6. Civil Liberties
7. Real GDP
8. Government
9. GINI Index

Below is some detail about GDP per capita and Inflation rate as these were the key factors that determine the country's status of prosperity.

1. GDP PER CAPITA: It is obtained by dividing the Real GDP with the country's overall population. In a word, it gives the average individual person output of a country.

2. INFLATION: It is the rate at which the consumer prices were rising in the country and the decline in the currency's purchasing power.

Note: It is not appropriate to use Real GDP along with above factors when predicting GDP Per Capita. Because they are same and doing so results is high Multicollinearity.

Now, that we have the 9 variables, we must find the variable which is dependent on remaining variables. In other words, finding a variable which is best explained by the remaining 8 variables.

Doing so we can get an overview of the data and decide on appropriate model to fit the data.

The reason why we are establishing a relationship and a model is, to predict the prosperity of some new countries which were not actually in our data set.

METHODOLOGY:

(i) Data Pre-processing:

- The Data set downloaded from CIA world Data records is cleaned for any missing values.
- The Independent variables that are chosen in an assumption that these variables are significant predictors of Dependent variable.
- These variables can be copied from various worksheets in the world data file into a new worksheet or new csv file.
- Since this project will be using multiple platforms like Python's Pandas, Sklearn, NumPy and SAS Studio for Data analysis and Model Fitting, a new CSV file is created which has the pre-processed data.
- Once the variables were gathered, the entire data must be sorted in such a way that the countries names were in A-Z format and so are their corresponding values like Inflation, GINI Index, GDP and so on. This can be done in Excel by deleting the countries names that were not matching with other independent variables
- Because as mentioned previously, that many independent variables can be taken from other worksheets and those variables has country

names that were not present in other variables. Hence in order to balance the dataset we must exclude the rows of the specific countries that were not present in other variables.

- The procedure is explained in Fig. 1 – Fig. 5

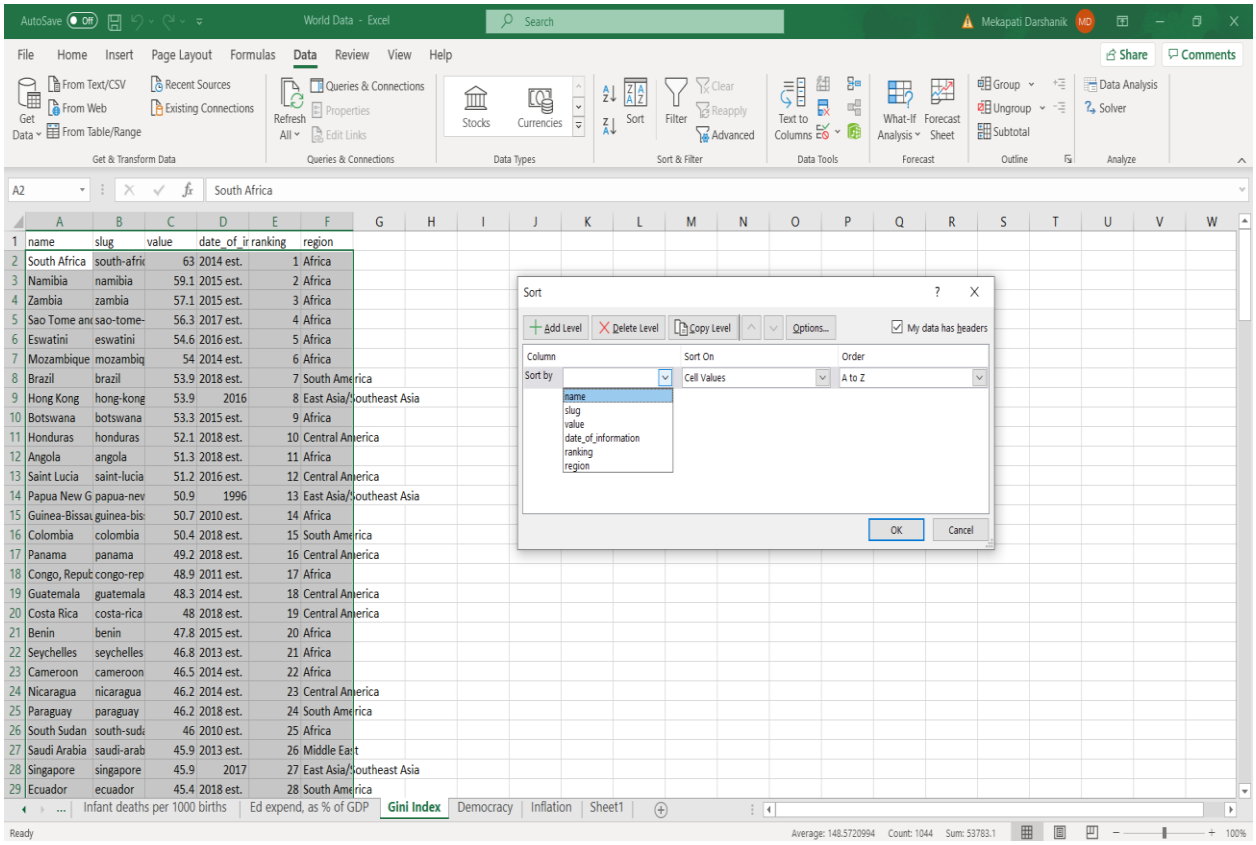


Fig 1. Data set before Sorting

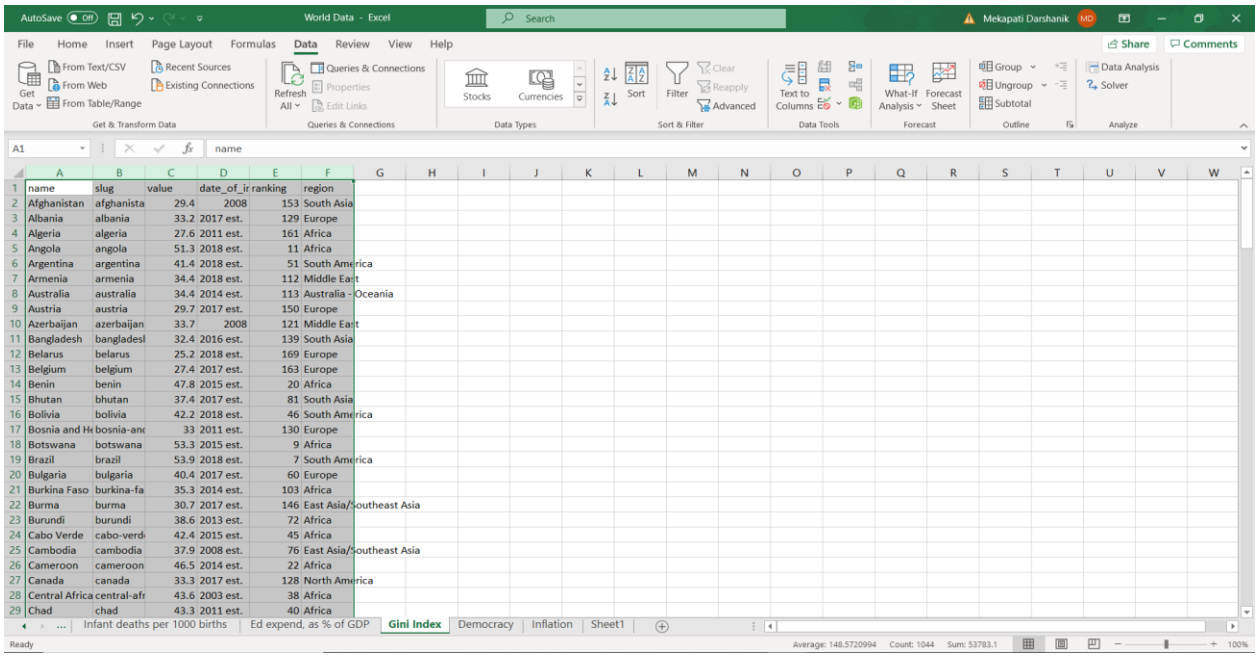


Fig 2. Data set After Sorting

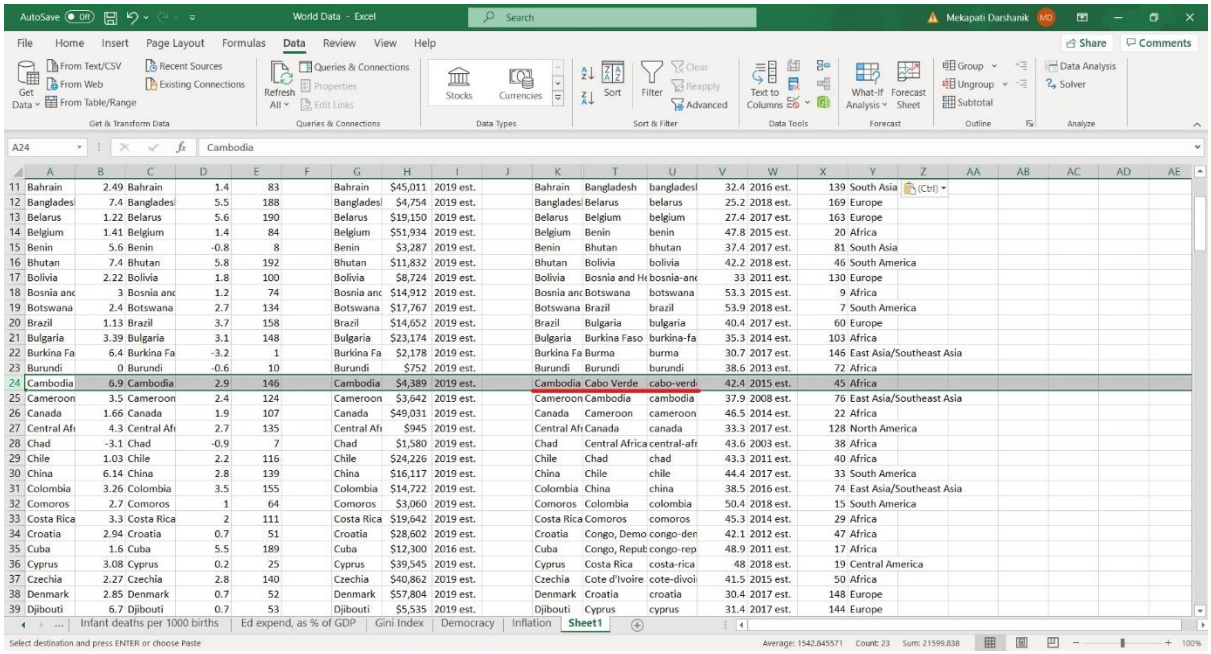


Fig 3. Imbalanced Data Set

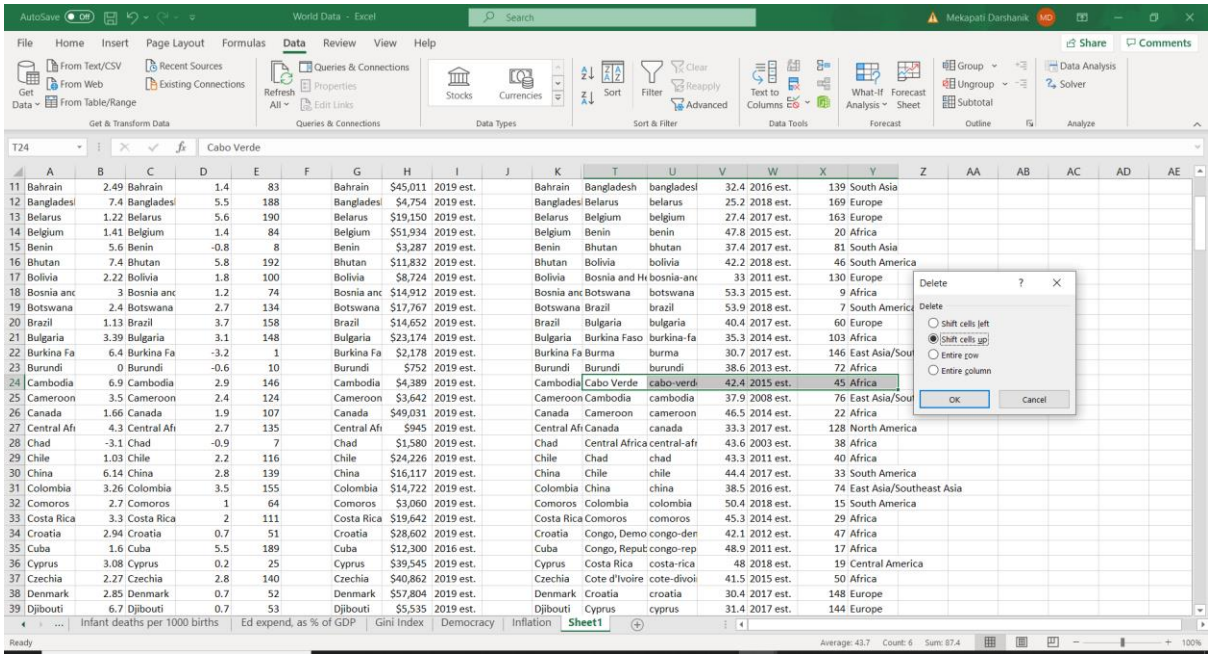


Fig 4. Dropping Imbalanced Rows

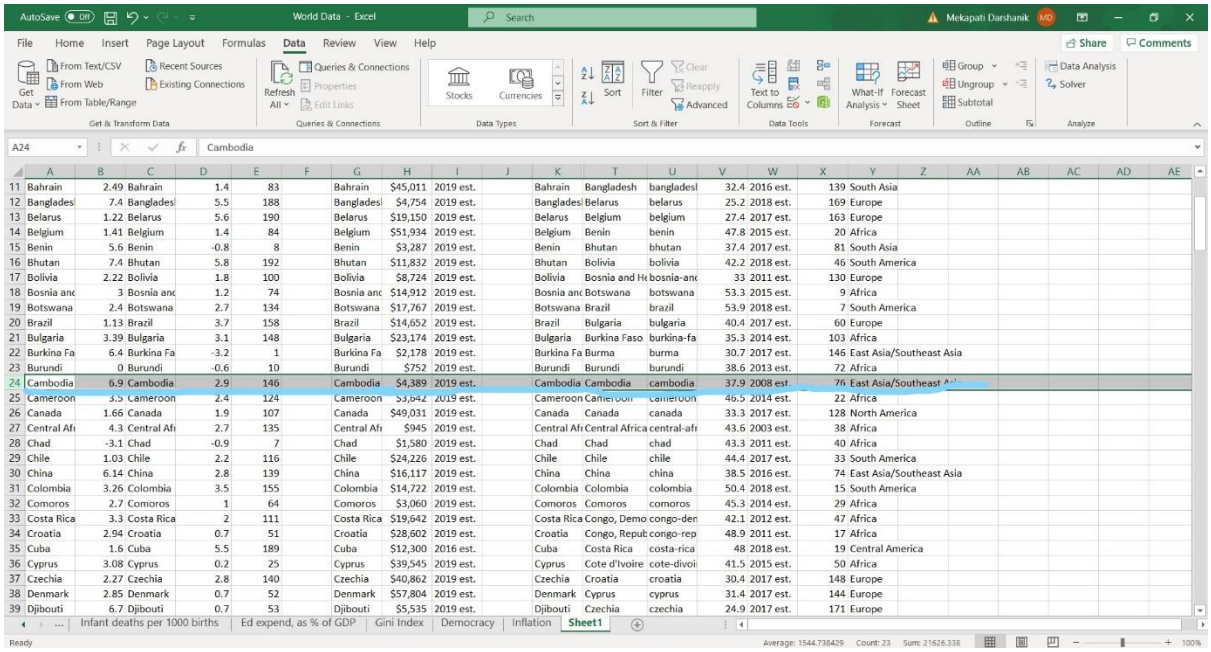


Fig 5. Balanced Data Set

(ii) Descriptive Statistics and Goodness of Fit:

Below is the Python's Stats models output of Data set by using ordinary least square method.

Table 1. Descriptive Statistics

	Gdp_Per_Capita	pluralism	government	politicalCulture	Population	Inflation	name	
count	150.000000	150.000000	150.000000	150.000000	1.500000e+02	150.000000	150	150.000000
unique	NaN	NaN	NaN	NaN	NaN	NaN	150	NaN
freq	NaN	NaN	NaN	NaN	NaN	NaN	1	NaN

	Gdp_Per_Capita	pluralism	government	politicalCulture	Population	Inflation	name	
mean	22367.106667	6.160533	5.072133	5.660867	4.919060e+04	3.826667	NaN	9.483270
std	21641.997901	3.540176	2.478235	1.680405	1.681354e+05	5.165030	NaN	1.142052
min	752.000000	0.000000	0.000000	1.880000	3.433530e+02	-3.200000	NaN	6.622736
25%	5443.500000	3.252500	3.105000	4.380000	4.143376e+03	0.900000	NaN	8.602131
50%	14478.000000	7.420000	5.360000	5.630000	1.016405e+04	2.350000	NaN	9.580372
75%	33152.000000	9.170000	7.052500	6.722500	3.393522e+04	4.475000	NaN	10.408852
max	114482.000000	10.000000	9.640000	10.000000	1.444216e+06	28.500000	NaN	11.648173

- The total number of data points in data set after pre-processing were 150 which is denoted as “Count” in the table 1 and similarly unique and frequency (denoted as freq) in table 1 depicts the unique and number of times the classification or non- numeric variable has appeared in the data set.
- The only non - numeric variable in our Data set is the “Name” column (or variable) which has the list of country names. Hence only that variable has unique values = 150 and rest other variables has NAN (Not A Number)
- If we look at mid and bottom section of table 1 it gives us some important insights of our Dataset such as the Spread of data points in each Variable by Standard deviation (denoted as std), Mean of variables, upper quartile, lower quartile and Median (or 50%) of the data set variables.
- For example, if we look at Gdp_Per_Capita variable, **up to or below 25% (or 1/4th or 1st Quarter) of the Data (in this case, Individual’s**

income in $) have GDP per Capita of $5443.50 and **up to 50% or below (Median) of the data have income of $ 14478.00** and lastly **up to 75% or below (Upper Quartile) have GDP Per Capita of $ 33152.00**

Goodness Of Fit.

- However, the more appropriate measure for Goodness of Fit is **Adjusted R squared** and to determine the Level of Significance of Independent variables in predicting the Dependent variable t-test values and P-values were determined by Python's Stats Models API.
- The P-values of government, political culture and Inflation are less than the α value of 0.05. Hence, the null hypothesis "There is no significant relationship between independent variables and dependent variables" is

rejected.

Out[13]:

OLS Regression Results

Dep. Variable:	Gdp_Per_Capita	**R-squared:**	0.428
Model:	OLS	**Adj. R-squared:**	0.413
Method:	Least Squares	**F-statistic:**	27.17
Date:	Sun, 01 Aug 2021	**Prob (F-statistic):**	7.84e-17
Time:	14:09:31	**Log-Likelihood:**	-1667.7
No. Observations:	150	**AIC:**	3345.
Df Residuals:	145	**BIC:**	3361.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1.426e+04	5255.161	-2.714	0.007	-2.47e+04	-3877.339
government	2770.1504	772.980	3.584	0.000	1242.387	4297.914
politicalCulture	4412.3064	1127.629	3.913	0.000	2183.593	6641.020
Inflation	-581.0629	273.393	-2.125	0.035	-1121.412	-40.714
Population	-0.0035	0.008	-0.424	0.672	-0.020	0.013

Omnibus:	58.986	**Durbin-Watson:**	2.092
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	163.063
Skew:	1.593	**Prob(JB):**	3.90e-36
Kurtosis:	6.992	**Cond. No.**	6.86e+05

Fig 6. Goodness of Fit and Level of Significance Test

- However, in Fig 6, under the Column P > |t| for Population variable the p-value is 0.672 > α value of 0.05. So, this variable is not a significant predictor for dependent variable at all levels of significance.
- Hence, the Population variable is dropped from the Data set as it is not significant predictor of GDP Per Capita.

Correlation Matrix and Choosing Dependent Variable:

- The correlation Matrix or simply called correlation describes the dependency of GDP per capita on all the existing independent variables.

- The closer the value to 1 the higher is the dependency.
- Fig 7 shows SAS Output of the correlation of GDP Per Capita with the 10 Independent variables.
- Almost all variables are having high correlation with GDP per Capita except population. However, Inflation and GINI Index (denoted as value) are negatively correlated with GDP Per Capita.
- On the other hand, Fig 8 shows the Python correlation coefficients of GDP per Capita with all remaining 8 variables (GINI Index is not considered in python). Similarly, almost all variables are having strong correlation with GDP Per Capita.
- Suppose if Inflation is considered as Dependent variable instead of GDP Per Capita, the Dependency of Inflation is not strong when compared and is show in Fig 9.
- Variables like political culture, political participation, GDP Per Capita and all variables that were previously showing high positive correlation are showing weak negative correlation if Inflation is the Dependent variable rather than GDP Per Capita.
- Hence this is the main reason why GDP Per Capita is considered as Dependent variable but not Inflation though both are good describers of Country's prosperity (Inflation will explain the prosperity in an inverse way though).

7/31/2021 Results: Correlation Analysis

1 With Variables:	Gdp_Per_Capita
10 Variables:	democracyScore pluralism government politicalParticipation politicalCulture civilLiberties Population Democratic_Status Inflation value

Pearson Correlation Coefficients
Number of Observations

	democracyScore	pluralism	government	politicalParticipation	politicalCulture	civilLiberties	Population	Democratic_Status	Inflation	value
Gdp_Per_Capita	0.56498 143	0.37468 143	0.61701 143	0.46397 143	0.61650 140	0.54582 143	-0.06602 143	0.49376 143	-0.27084 143	-0.28269 143

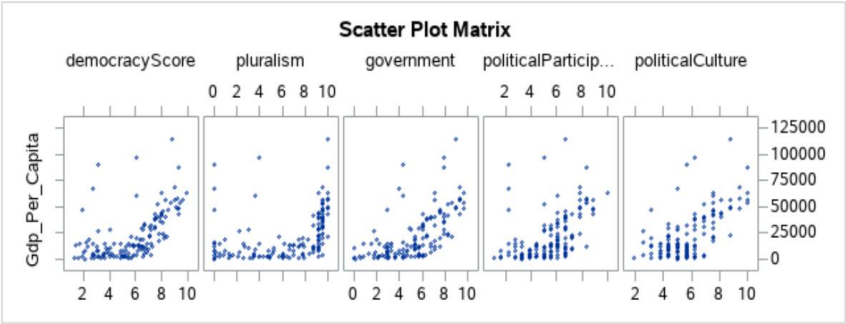
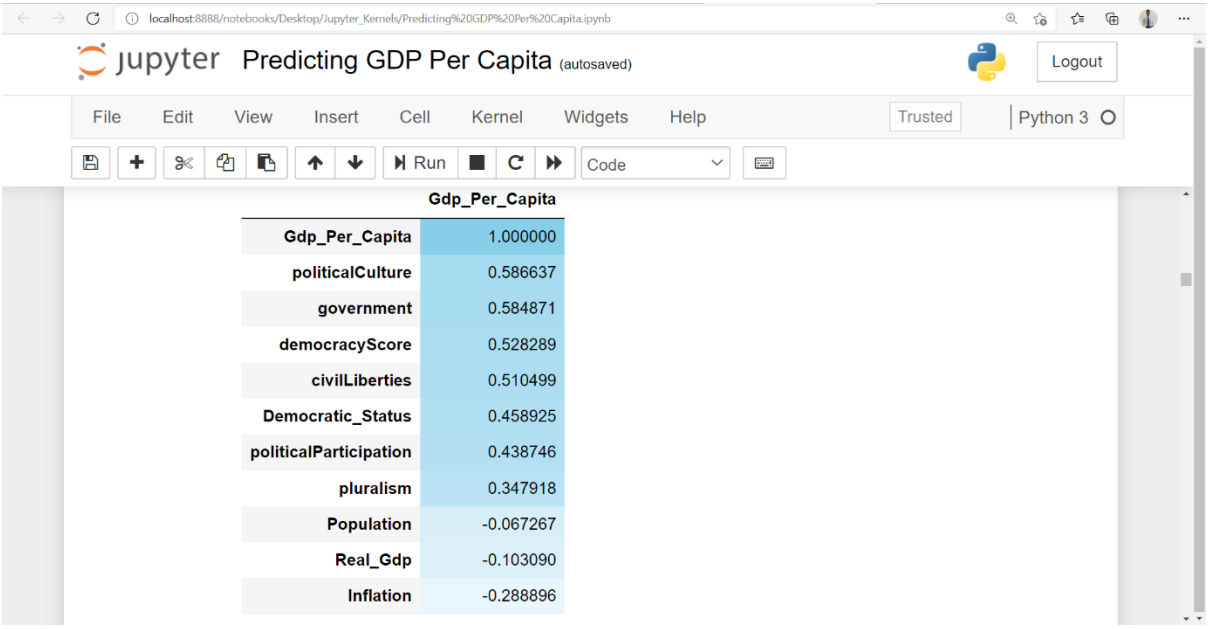
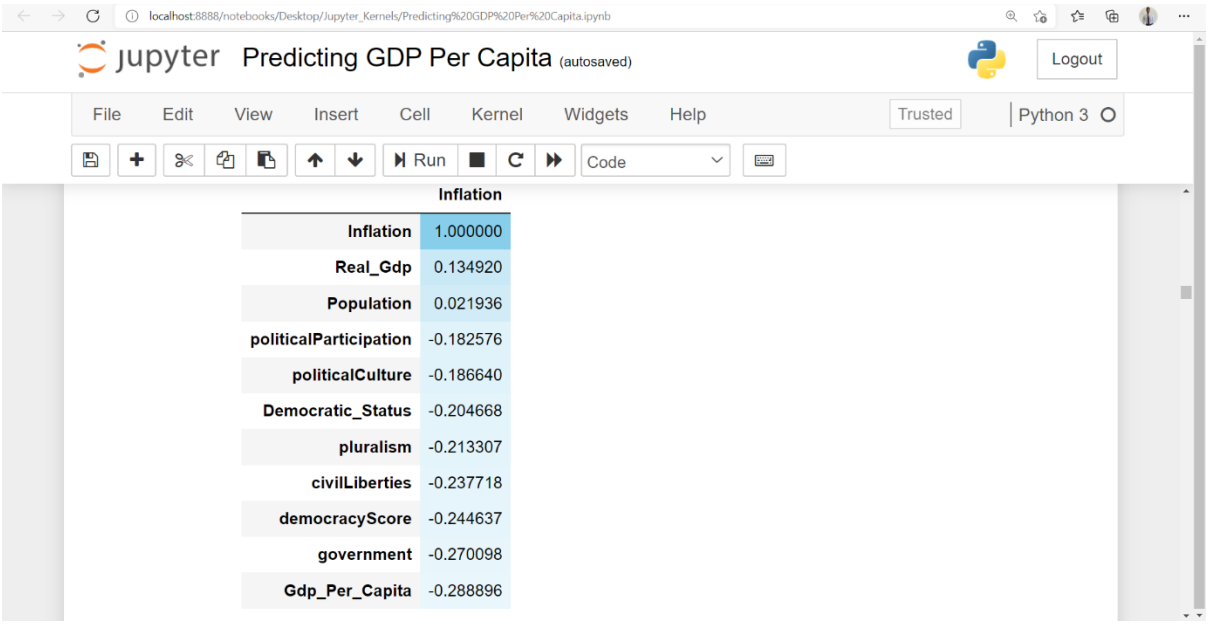


Fig 7.SAS Output of Correlation



	Gdp_Per_Capita
Gdp_Per_Capita	1.000000
politicalCulture	0.586637
government	0.584871
democracyScore	0.528289
civilLiberties	0.510499
Democratic_Status	0.458925
politicalParticipation	0.438746
pluralism	0.347918
Population	-0.067267
Real_Gdp	-0.103090
Inflation	-0.288896

Fig 8. Correlation if GDP Per Capita is Dependent variable



	Inflation
Inflation	1.000000
Real_Gdp	0.134920
Population	0.021936
politicalParticipation	-0.182576
politicalCulture	-0.186640
Democratic_Status	-0.204668
pluralism	-0.213307
civilLiberties	-0.237718
democracyScore	-0.244637
government	-0.270098
Gdp_Per_Capita	-0.288896

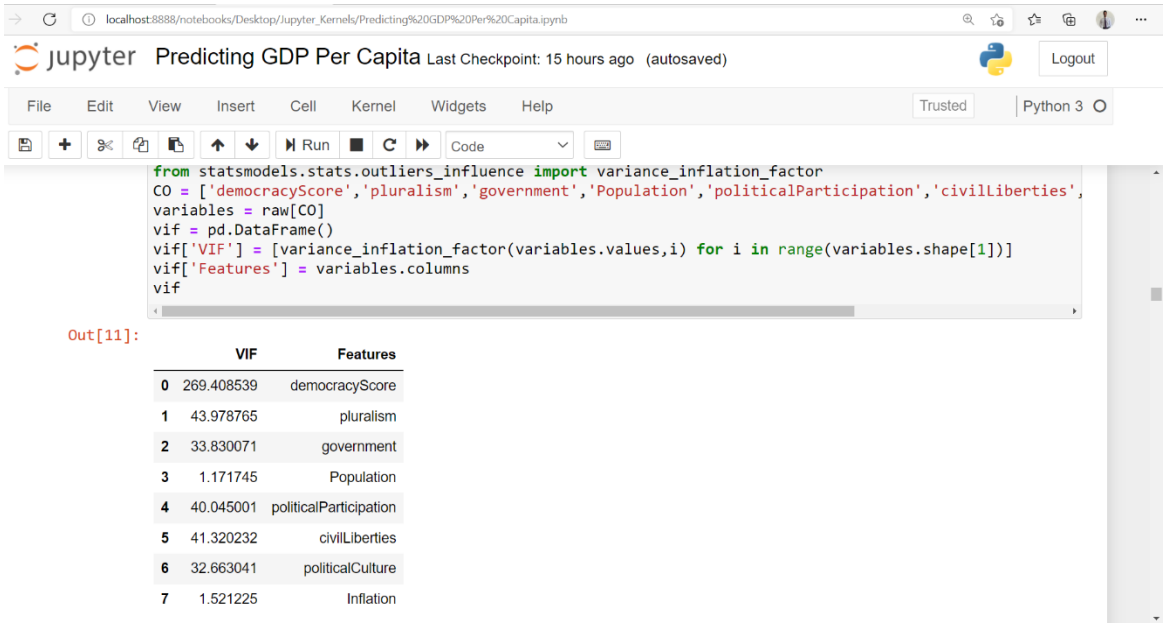
Fig 9. Correlation if Inflation is Dependent variable

Multicollinearity:

- Though the Adjusted R squared gives the appropriate measure for goodness of fit by reducing the over fitting of model and correlation specifies the explaining power of independent variables, the existing

independent variables may have high dependency among themselves apart from that of Dependent variable “GDP Per Capita”. This type of inter dependency among the independent variables is called Multicollinearity.

- Variance Inflation factor (VIF) measures the Multicollinearity of Independent variables.
- Any Independent variable whose VIF is greater than 10-20 range are highly Multicollinear with other Independent Variables.
- Fig 10. Shows the VIF of all Independent variables and **Democracy score has very high VIF value (269.408)** which clearly states that the variable is contributing to High multicollinearity which ultimately leads to over fitting of Model. Similarly, Pluralism and civil liberties are having the next highest VIF value of 43.97 and 41.32 respectively.



	VIF	Features
0	269.408539	democracyScore
1	43.978765	pluralism
2	33.830071	government
3	1.171745	Population
4	40.045001	politicalParticipation
5	41.320232	civilLiberties
6	32.663041	politicalCulture
7	1.521225	Inflation

Fig 10. Variance Inflation Factor

- The variables like Democracy score and political participation are dropped from the Data set because it is having abnormal multicollinearity.

```
In [14]: def MUL(c):
             from statsmodels.stats.outliers_influence import variance_inflation_factor
             variables = raw[c]
             vif = pd.DataFrame()
             vif['VIF'] = [variance_inflation_factor(variables.values,i) for i in range(v
             vif['Features'] = variables.columns
             print(vif)
         MUL(['pluralism','government','politicalCulture','civilLiberties','Inflation'])
```

	VIF	Features
0	21.071283	pluralism
1	20.766281	government
2	14.696061	politicalCulture
3	37.444850	civilLiberties
4	1.460932	Inflation

Fig 11. VIF after dropping democracy score and political participation

- From Fig 11, only civil liberties is having very high VIF and it is not considered for modelling similar to that of democracy score and political participation.

(iii) Distribution and Log Transformation

- Since the Dependent variable GDP Per Capita has data that are in currency format, this will add more weights(coefficients) to this variable when fitting a model. This will ultimately yield in overfitted model.
- To avoid overfitting and biasing in allocating the weights of the variable, the data is converted to its natural logarithm so that the dependent variable will be in the appropriate scale of its independent variable.
- Moreover, doing so will also display the dispersion of the data in Dependent variable and will normalize the data if it is +vely or -vely skewed as shown in Fig 12 and 13. The Blue curve is the distribution of dataset and Black curve is the ideal curve of normal distribution.

```
ax1 = plt.subplot2grid((2,1),(1,0))
plt.title('Probability distribution of Gdp Per Capita(Y-Variable) After Transform', fontsize = 15)
sns.distplot(targets, fit = norm)
```

```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x2202cdac7c8>
```

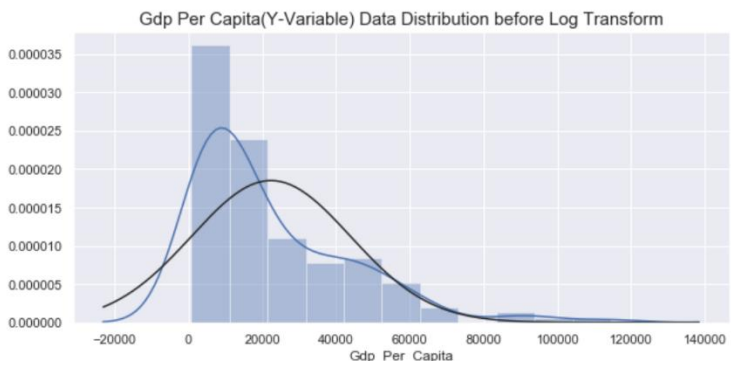


Fig 12. Shape of Data in GDP Per Capita before Log Transform

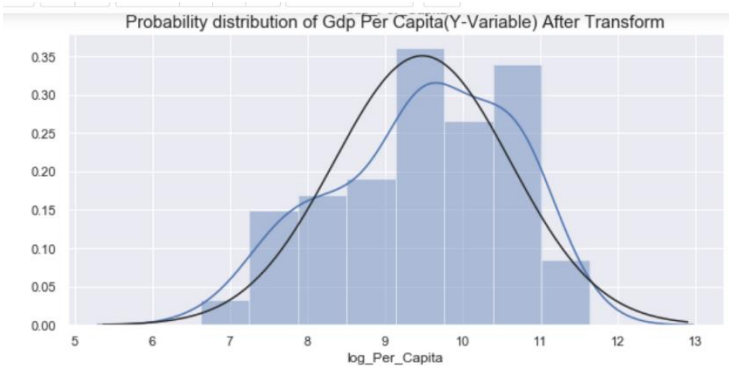


Fig 13. Approx. Normal Distribution of Data in GDP Per Capita After Log Transform

(iv) Pattern And Outlier Detection:

- Fig 14.1 and 14.2 shows the quick Visualization of scatter plot matrix among all the variables in the data set. GDP Per Capita is having a linear distribution of data democracy score, pluralism, government and civil liberties.
- But despite having close to linear data distribution of democracy score, civil liberties, and political participation they are having high multicollinearity as mentioned in above section. Hence these were not considered as strong predictors for GDP Per capita even though they satisfy the p-value < 0.05 criterion.

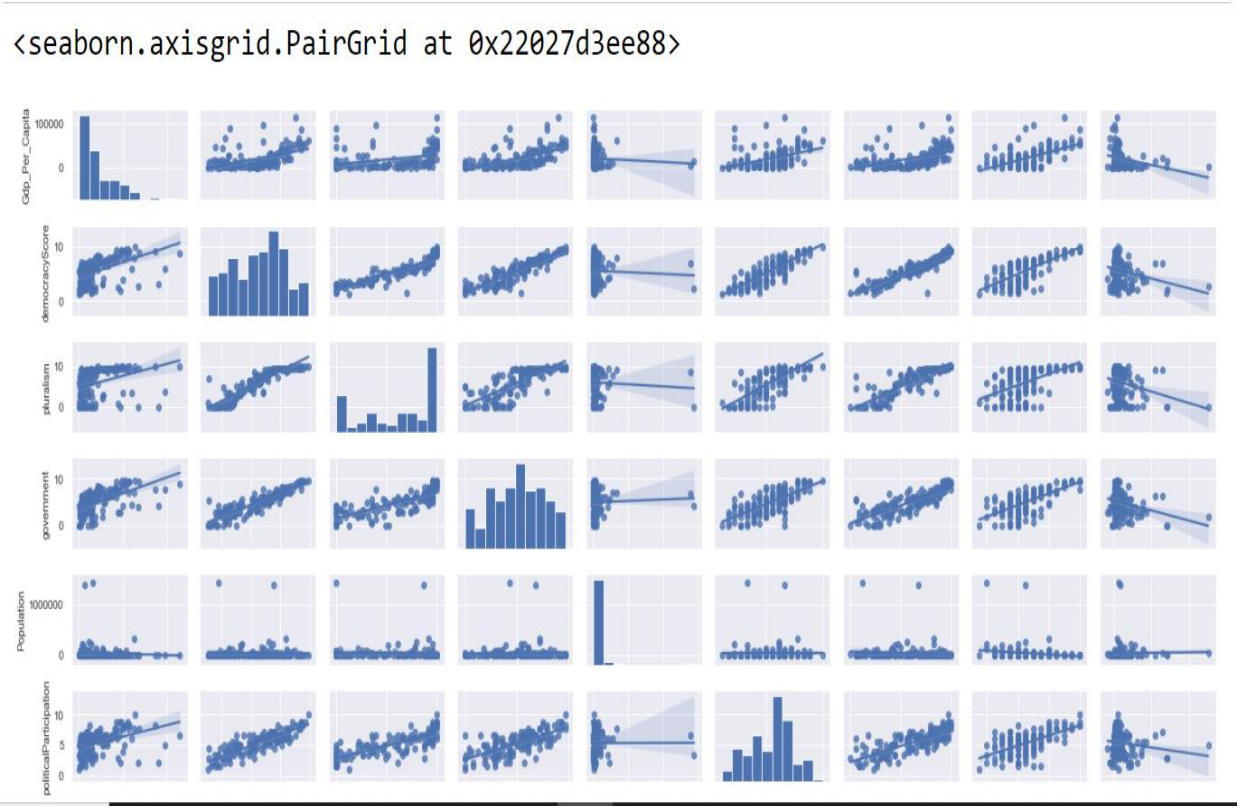


Fig 14.1. Scatter plot among variables(a)

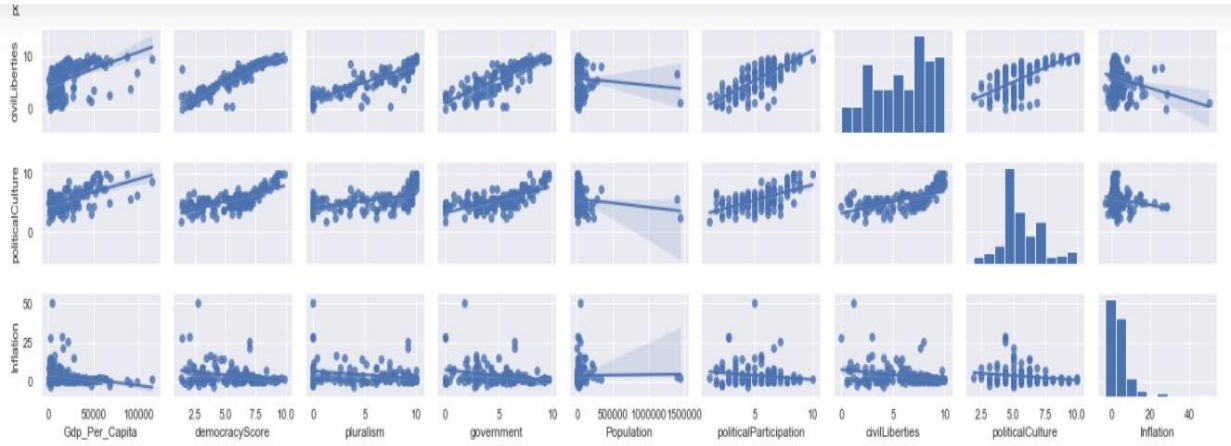


Fig 14.2. Scatter plot among variables(b)

Outliers:

- Any data point falling above upper quartile (Q3 or 75% of the data) or below the lower quartile (Q1 or 25% of data) of a variable is called an outlier

- Fig 15 shows the Box and whisker plots of the Pre-processed data set which is free from multicollinear variables and missing values.
- Of all the variables only Population variable has outliers, and they are ranging from 200000 to 1400000.
- Hence the Population variable is also dropped from the data set so that the model does not over fit.
- Finally, the data set is left with only 4 variables they are Inflation, political culture, government, and pluralism.

Fig 15. Box and Whisker plot and Outliers

(v) Model Fitting, Training and Evaluation.

- Now the Data set is cleaned and free from any outliers and multicollinear variables, it is ready for model building.
- Initially, the data set is having 150 rows and 5 columns of which one column is the dependent variable (GDP Per Capita)
- The data set is divided into training and validation sets so that the model is trained on the training data set and steps are taken to fit the model to the training set as close as possible. Once the model is fitted to trained data set, it is given a new set which is stored in validation after each consecutive prediction of the training data set.
- This will result in refining the model performance as it is trained and also tested or validated at same time and there by yielding in lower RMSE value.
- The size of trained set is 120 rows(80% of original data set) and size of testing or validation data set is 30 rows(20% of original data set).

- Python's Sklearn library is used to perform regression and also its enhanced models like XG Boost and stacking regression.
- The RMSE of Linear regression model is 0.56389 or 56.389% which is not great as the model is only 56% reliable which is shown in Fig 16.
- For the model to fit more accurately in the data set, XG Boost Algorithm is used rather than following conventional method of dropping the variables which will compromise the R squared of the model or adding the variables which will increase the Multicollinearity among independent variables.

```
In [30]: from sklearn.metrics import mean_squared_error
         MSE = cross_val_score(linreg, x_train, y_train, scoring = 'neg_mean_squared_error', cv = 5)
         mean_MSE = np.mean(MSE)

         ## Here neg_mean_squared error is selected as reference from sklearn website(Appropriate for regressior

         print(mean_MSE)
         print('\n RMSE : ', np.sqrt(-mean_MSE))
```

```
-0.3230701823788678

 RMSE :  0.5683926304755084
```

Fig 16. Linear Regression Model and RMSE Score

- Regularization models like Ridge Regression are used to avoid overfit of model while getting trained on the data set as a preventive measure. This model returns the best possible RMSE score which is shown in Fig 17
- Finally, Fig 17 shows the **RMSE value of XG BOOST (Xtreme Gradient Boosting algorithm) as 0.092377 shown in Fig 18**and Ensemble Stacking Regression Algorithm as 0.09089 or 91% accuracy there by concluding the best model to predict the GDP Per Capita of Countries.

```
In [32]: ridge_mod = Ridge(alpha = 4)
         ridge_mod.fit(x_train,y_train)
         y_pred_train = ridge_mod.predict(x_train)
         y_pred_test = ridge_mod.predict(x_test)

         print('\RMSE: ', np.sqrt(mean_squared_error(y_train, y_pred_train)))
         print('\n RMSE : ', np.sqrt(mean_squared_error(y_test, y_pred_test)))

         \RMSE:  0.49673991764804126

          RMSE :  0.6103857079720303

In [33]: from xgboost.sklearn import XGBRegressor
         xgb = XGBRegressor(n_estimators = 5000, verbosity = 1)
         xg_reg = xgb.fit(x_train,y_train)
         xg_predict = xg_reg.predict(x_test)

         print('\n RMSE : ', np.sqrt(mean_squared_error(y_test,xg_predict)))

          RMSE :  0.09237734518581742
```

Fig 17. Ridge regularization, XG Boost model and RMSE Score

```
In [35]: from mlxtend.regressor import StackingRegressor

         st_reg = StackingRegressor(regressors = [ridge_mod], meta_regressor = xgb, use_features_in_secondary =

         stack_mod = st_reg.fit(x_train,y_train)
         stack_predict = stack_mod.predict(x_test)

         print('\n RMSE :', np.sqrt(mean_squared_error(y_test, stack_predict)))

          RMSE : 0.09089117856259
```

Fig 18. Ensemble Stacking Model and RMSE Score

- Fig 19 shows the Scatter plot and fitted XG Boost regression line reinforcing the RMSE score obtained in Fig 18. Fig 20 shows the Scatter plot between predicted and actual Log GDP Per Capita values for Ridge Regularization and Fig 21 showing the scatter plot for linear regression.

```
RMSE : 0.09089117856259
```

```
In [37]: verify_fit(y_test,stack_predict)
```

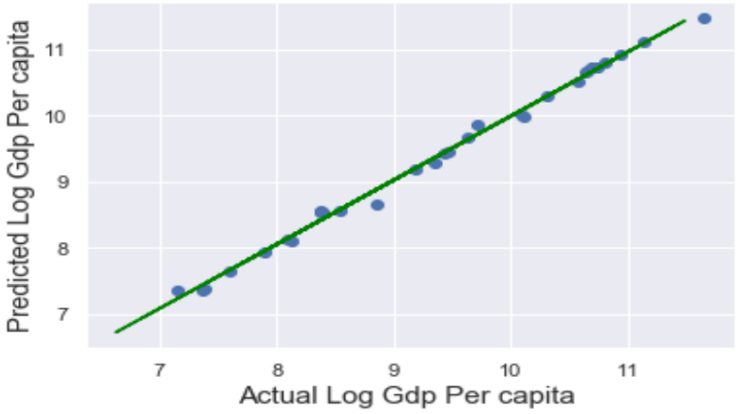


Fig 19. Ensemble Stacking Model Scatter plot

```
RMSE :  0.09237734518581742
```

```
In [35]: verify_fit(y_train,y_pred_train)
```

Fig 20. Ridge Model Scatter plot

```
In [29]: def verify_fit(x,y):
             plt.scatter(x,y)
             plt.xlabel('Actual Log Gdp Per capita ',fontsize = 15)
             plt.ylabel('Predicted Log Gdp Per capita',fontsize = 15)
             m, b = np.polyfit(x, y, 1)
             plt.plot(y_train, m*y_train+b, color = 'green')
         verify_fit(y_train,y_pred_train)
```

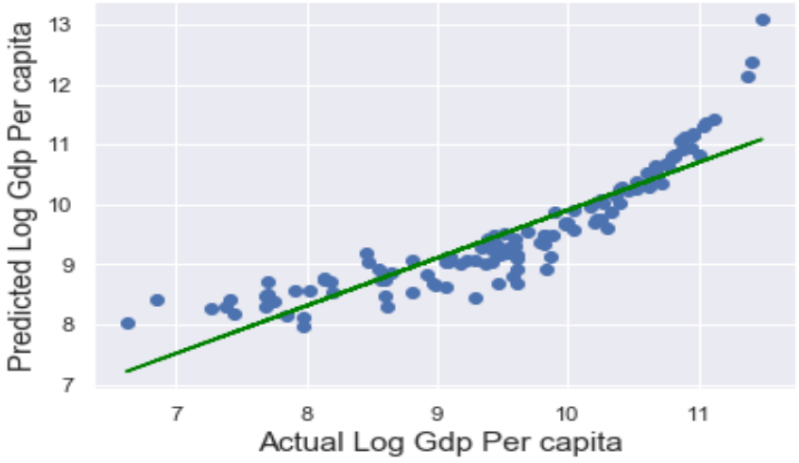


Fig 21. Linear Regression Model Scatter plot

Table 2. Ensemble Stacking Regression Predictions of GDP Per Capita in $

15874.588 , 2807.1707, 29736.936 , 1594.3081, 12724.341 ,	15874.588 , 2807.1707, 29736.936 , 1594.3081, 12724.341 ,	15874.588 , 2807.1707, 29736.936 , 1594.3081, 12724.341 ,	15874.588 , 2807.1707, 29736.936 , 1594.3081, 12724.341 ,	15874.588 , 2807.1707, 29736.936 , 1594.3081, 12724.341 ,
2088.8494, 10881.557 , 43553.508 , 1575.5475, 96410.54 ,	2088.8494, 10881.557 , 43553.508 , 1575.5475, 96410.54 ,	2088.8494, 10881.557 , 43553.508 , 1575.5475, 96410.54 ,	2088.8494, 10881.557 , 43553.508 , 1575.5475, 96410.54 ,	2088.8494, 10881.557 , 43553.508 , 1575.5475, 96410.54 ,
12500.121 , 3284.603 , 5278.573 , 5069.117 , 22250.557 ,	12500.121 , 3284.603 , 5278.573 , 5069.117 , 22250.557 ,	12500.121 , 3284.603 , 5278.573 , 5069.117 , 22250.557 ,	12500.121 , 3284.603 , 5278.573 , 5069.117 , 22250.557 ,	12500.121 , 3284.603 , 5278.573 , 5069.117 , 22250.557 ,
49184.332 , 37322.383 , 68328.29 , 5149.6167, 3361.5107,	49184.332 , 37322.383 , 68328.29 , 5149.6167, 3361.5107,	49184.332 , 37322.383 , 68328.29 , 5149.6167, 3361.5107,	49184.332 , 37322.383 , 68328.29 , 5149.6167, 3361.5107,	49184.332 , 37322.383 , 68328.29 , 5149.6167, 3361.5107,
9734.11 , 19118.83 ,	9734.11 , 19118.83 ,	9734.11 , 19118.83 ,	9734.11 , 19118.83 ,	9734.11 , 19118.83 ,

46464.316 , 42565.37 , 45989.348 ,	46464.316 , 42565.37 , 45989.348 ,	46464.316 , 42565.37 , 45989.348 ,	46464.316 , 42565.37 , 45989.348 ,	46464.316 , 42565.37 , 45989.348 ,
5829.284 , 5209.1763, 1567.8215, 56370.383 , 21725.86	5829.284 , 5209.1763, 1567.8215, 56370.383 , 21725.86	5829.284 , 5209.1763, 1567.8215, 56370.383 , 21725.86	5829.284 , 5209.1763, 1567.8215, 56370.383 , 21725.86	5829.284 , 5209.1763, 1567.8215, 56370.383 , 21725.86

RESULTS AND CAVEATS:

- Fig 22 shows the SAS output of Linear Regression Model and its adjusted R squared values. Analogically, they are similar to OLS results obtained from Python. This further solidifies the Goodness of Fit measure for Multiple Linear Regression Model.
- Fig 23 shows the Neural Networks RMSE score. Microsoft Azure Cloud and Machine Learning classic studio is used for creating the experiment tree of neural networks. This is similar to SAS and is automated cloud based Machine learning solution developed by Microsoft.
- The reason why neural networks is also used in predicting the GDP Per Capita is because the SAS's Rapid predictive Modeler has provided Neural networks as one of the model comparisons along with Linear regression and Ensemble estimates.
- However, only Ensemble estimate has provided valid RMSE score compared to Linear Regression and even Neural networks.

Reason For Low performance of Neural networks:

- Neural network is an excellent model for predicting almost any type of data set. **But the main limitation is the data set should be big in size (greater than 1000 rows and 100 columns).**
- **It is not suitable for smaller data sets like the one used in this project (150 rows and 10 columns).**

7/31/2021 Results: Linear Regression

Model: MODEL1
Dependent Variable: Gdp_Per_Capita

Number of Observations Read	144
Number of Observations Used	140
Number of Observations with Missing Values	4

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	32777572911	8194393228	31.89	<.0001
Error	135	34690898652	256969620		
Corrected Total	139	67468471563			

Root MSE	16030	R-Square	0.4858
Dependent Mean	22426	Adj R-Sq	0.4706
Coeff Var	71.48009		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-1666.64584	9747.97756	-0.17	0.8645
government	1	2951.39488	779.31642	3.79	0.0002
politicalCulture	1	4296.83786	1112.19332	3.86	0.0002
Inflation	1	-858.74974	338.63344	-2.54	0.0124
value	1	-342.79262	187.03633	-1.83	0.0690

Model: MODEL1
Dependent Variable: Gdp_Per_Capita

Fig 22. SAS Output for Goodness of Fit and Level of Significance

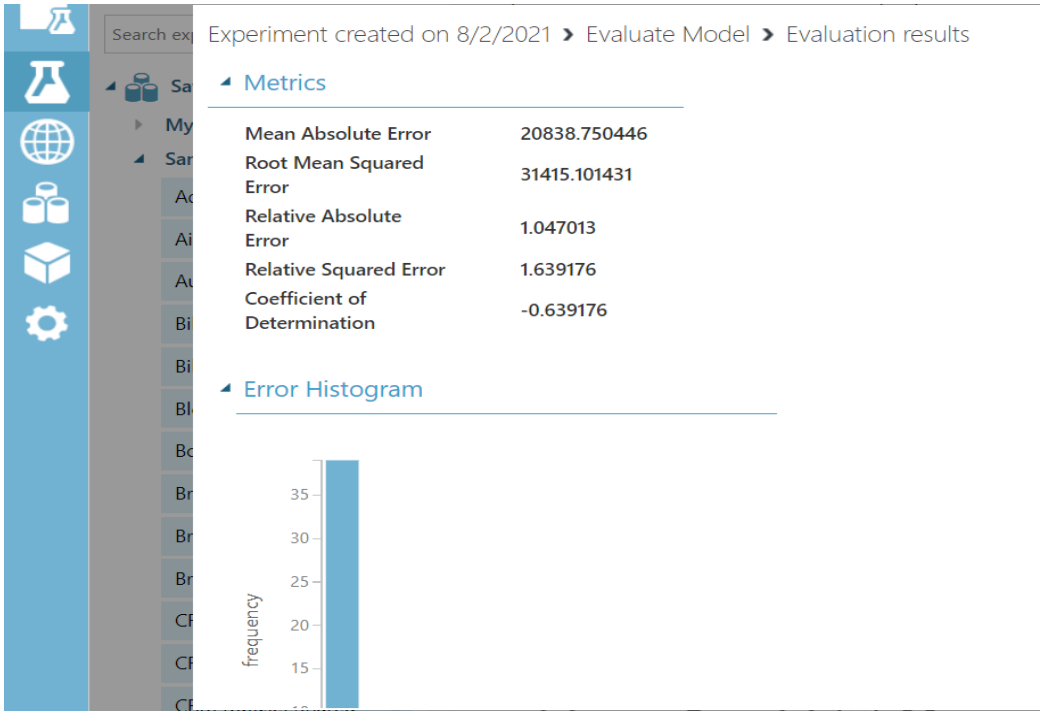


Fig 23. Azure's Neural networks RMSE Score

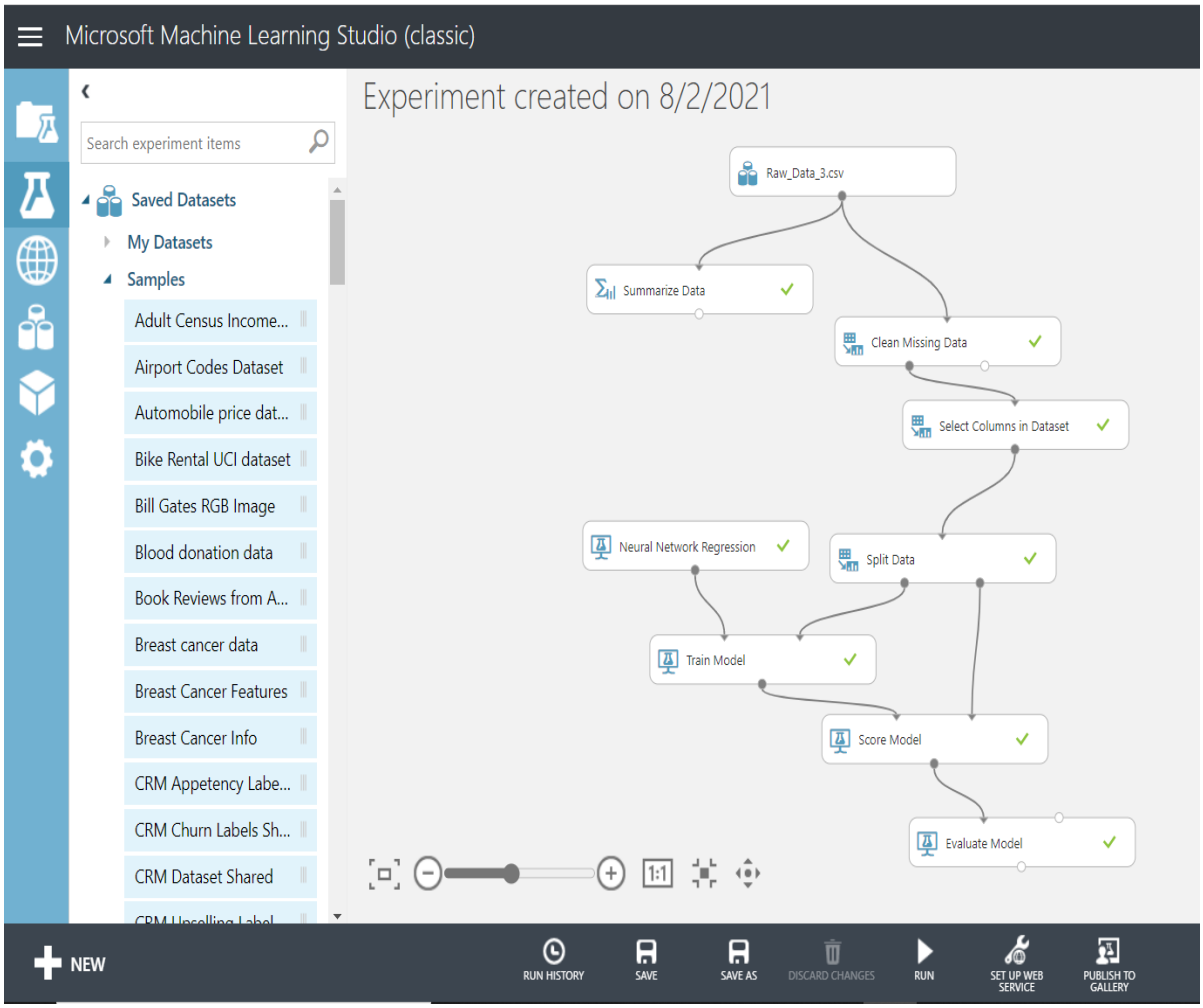


Fig 24. Azure's Neural networks Experiment Tree

CONCLUSION:

- Finally, based on the RMSE Scores and Adjusted R squared for various regression models and ensemble estimates, The Ensemble Estimate's Stacking Regression Model is the Best in determining the GDP Per Capita of countries and ultimately predicting the Country's prosperity.
- Moreover, the Data set with which we initially started has 9 columns(variables) but if we notice the data set that we trained the model has only 5 columns. The main reason for reduction in Data set size is due to Severe Outliers and multicollinearity among independent variables.

SUPPORTING MATERIALS

- Refer PDF “GINI Index and GDP Per Capita” for Rapid Predictive Modelling output.
- Refer PDF “Python Code for Predicting GDP” for code, data analysis and Model fitting and evaluation.

FURTHER IMPROVEMENTS:

- Because neural networks work on nodes of hidden layers and backpropagation method to perform regression at each node and correct the error the node has made by using back propagation similar to the way of how a human brain works and learns the new things.
- A simple neural network has an “**Input layer, output layer and one or more hidden layers**”. Each layer in network has nodes depicting the variables in the data set and all the nodes are interconnected with each other. So that the error done by one node is rectified by the succeeding node or layer.
- However, different and positive results with lower RMSE are possible even for small data sets such as the one used in this project. But Python’s Tensor flow and Keras might be used to manually code the neural network and tune the network by appropriate backpropagation techniques readily available.
- This segment of neural network development using Tensor flow and keras can be the source for the continuation and further improvement of this project thereby by utilizing the maximum capability of neural network.