# Project Report On

# Economic Growth: A Machine LearningApproach To Gdp Per Capita Prediction

**INTRODUCTION**

- **Overview**

Gross Domestic Product is cited as a vital and most widely accepted economic indicator which not only helps in diagnosing the problems related to the economy but also correcting it. The usage of the gross domestic product as a measure of the market price of ultimate services and products that are produced over a selected amount of time will definitely continue to owe an abundance to the producing age. To policy makers and statisticians especially, gross domestic product helps in conveying data about the economy in particular and thereby notifying about a country's economic health. This paper makes an attempt to expedite the process of prediction ofGross Domestic Product.

- **Purpose**

We are here proposing a system that will be based on Decision Tree, Random Forest and BootStrap algorithms and will forecast the GDP of the Nation. We will use data from AgricultureProduction, Manufacturing Industry and Service Industry. These are the sectors of the main mainstream that are responsible for the effect on GDP forecasting. Also we will use Purchasing Power Parity as an input element for better prediction of GDP of a nation. We will use multiple algorithms for increasing and improving accuracy.

- **LITERATURE SURVEY**

- **Existing Problem**

Currently there is no automatic system that will predict GDP of whole country. There experts appointed by Government to study the information regarding multiple sectors of financial income providers. Currently it is critical work to analyze every field of industry running in country. Huge human work power need for this work. Again it doesn't guarantee accuracy of work results and output. GDP which is very most important index for any country in terms of planning of various things. Wrong or miss leading decision can effect on GDP index. It is hard tothink in and cover multiple aspects at same time for human.

- **Proposed Solution**

We are here proposing a system that will be based on Decision Tree, Random Forest and BootStrap algorithms and will forecast the GDP of the Nation. We will use data from AgricultureProduction, Manufacturing Industry and Service Industry. These are the sectors of the stream thatare responsible for the effect on GDP forecasting. Also we will use Purchasing Power Parity as an input element for better prediction of GDP of a nation. We will use multiple algorithms for increasing and improving accuracy.

Advantages of proposed system

• It would help in predicting the GDP of our country.

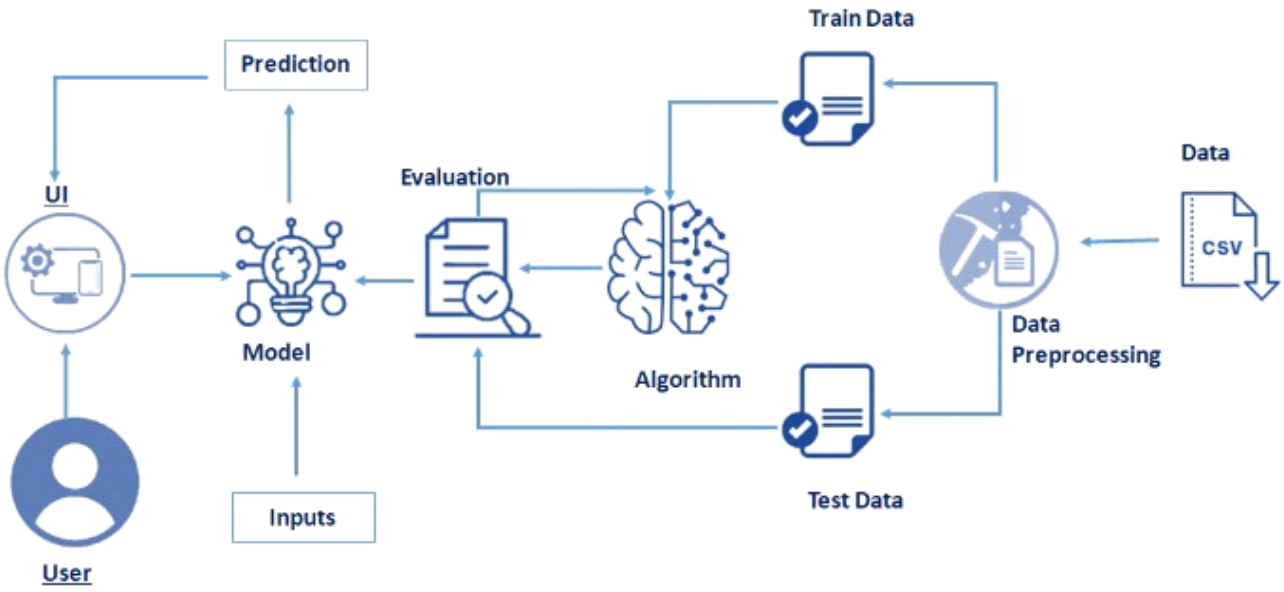
• It would help in making financial decisions and investments decisions with much ease.

• It would help in career guidance indirectly or directly.

• It would be of low cost and highly user friendly.

• It would not require much high maintenance until the parameters change.

- **THEORETICAL ANALYSIS**




- Hardware/Software Designing

1.Software Requirements

Downloading of Anaconda

Navigator2.Importing python

packages like

- NumPy Package
- Pandas
- seaborn
- joblib
- Matplotlib
- scikit-learn
- Flask

- LabelEncoder
- train_test_split

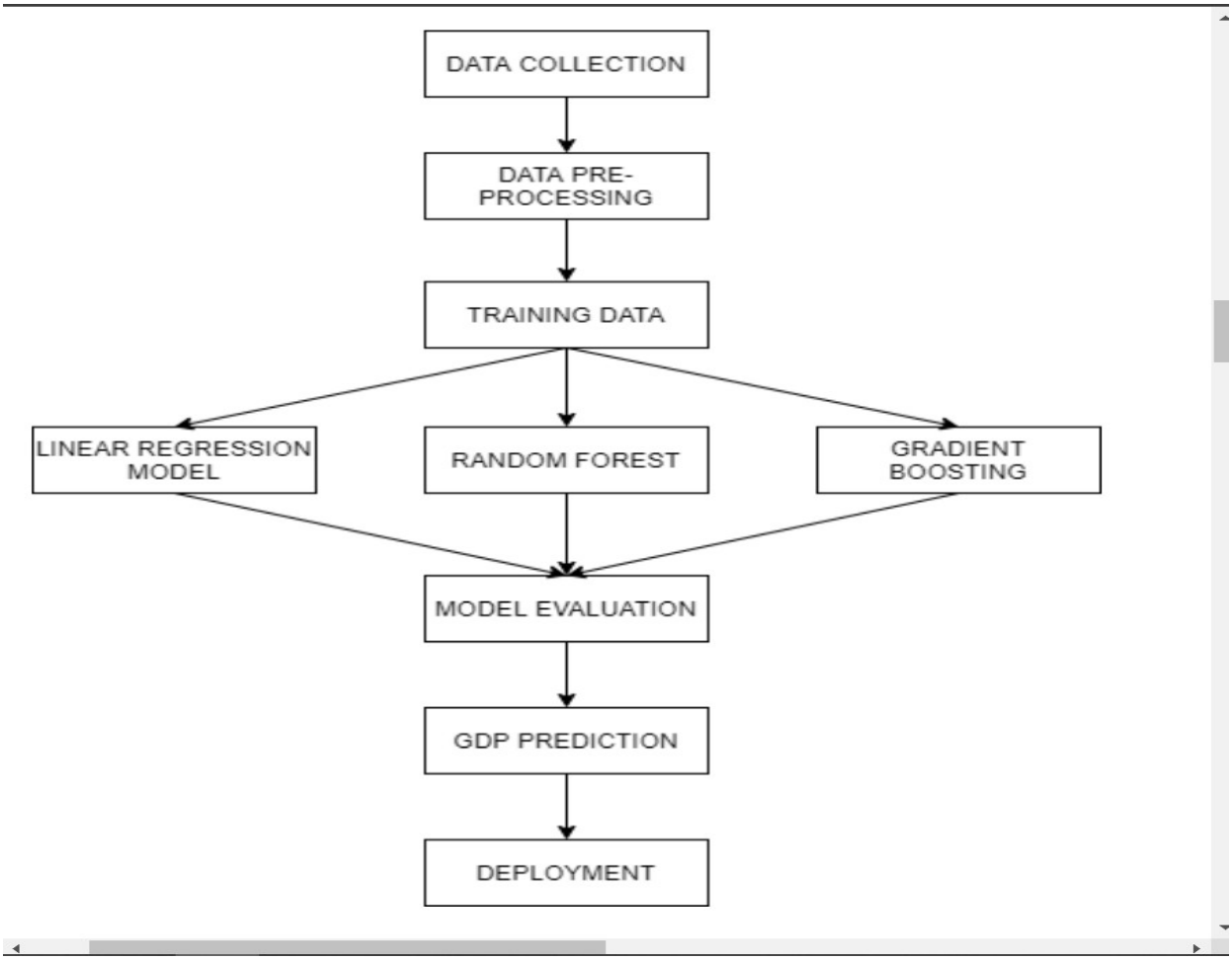
j. LinearRegression

k. RandomForestRegressor

- **EXPERIMENTAL INVESTIGATION**

We have imported many libraries. The Python Data Analysis Library (Pandas) is an acronym for"Python Data Analysis Library." Its will be used as the structure for doing reasonable, genuine information investigation in Python. Pandas are built on top of numpy, a package that supports multi-dimensional arrays. Many of the time-consuming, repetitive tasks associated with workingwith data are made easy with Pandas, including: Data cleaning, Data fill, Data normalization andJoins and merges. Numerical Python (NumPy) is used to solve problems numerically. It also hasfunctions for dealing with matrices and the domain of linear algebra. Benefits of using numpy are fast, fewer loops, cleaner code and improved quality. Seaborn is a matplotlib-based Python data visualization library. It has a high-level interface for producing visually pleasing and insightful statistical graphics. Matplotlib is a Python library that allows you to construct static, animated, and interactive visualizations. With only a few lines of code, you can create publication-quality plots.

- **FLOWCHART**




- **RESULT**

Fig:1 output of the dataset:

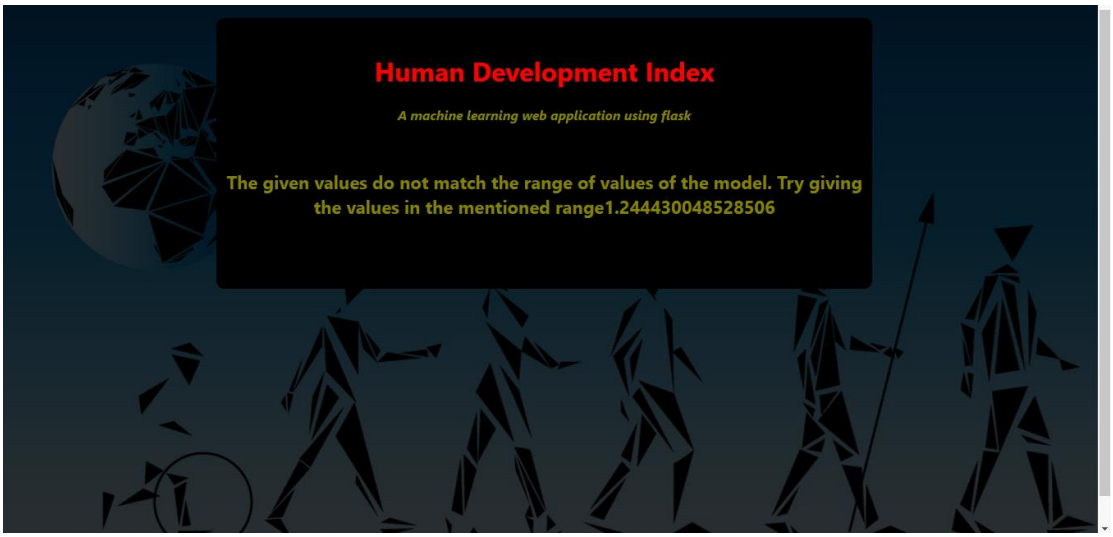


Fig 2: Web Application view :


Fig 3: Predicting the GDP:


- **ADVANTAGES AND DISADVANTAGES**

Advantages

- It reduces overfitting in decision trees and helps to improve the accuracy
- It is flexible to both classification and regression problems
- It works well with both categorical and continuous values
- It automates missing values present in the data

- Normalizing of data is not required as it uses a rule-based

approach.Disadvantages

- It requires much computational power as well as resources as it builds numerous trees tocombine their outputs.
- It also requires much time for training as it combines a lot of decision trees to determinethe class.
- Due to the ensemble of decision trees, it also suffers interpretability and fails to determinethe significance of each variable.

- **APPLICATIONS**

The accuracy obtained by linear regression algorithm is 82% and by random forest is 87%. On the basis of the optimization process, the machine learning algorithm "Gradient Boosting" utilized during this project worked well with the accuracy 89% in order to predict the true GDPper capita.

## 9.CONCLUSION

We explored all the supervised regression models in order to get the best fitting models. We havetrained the model using Linear Regression, Random Forest and Gradient Boosting machine learning algorithms and also estimated the performance of these models. Evaluation is done using MAE and RMSE techniques and then compared all three models to get a clear overview ofperformance. The accuracy obtained by linear regression algorithm is 82% and by random forestis 87%. On the basis of the optimization process, the machine learning algorithm "Gradient Boosting" utilized during this project worked well with the accuracy 89% in order to predict the true GDP per capita. Finally deployed the highest accuracy model to "GDP Estimation Tool" which estimates and forecasts GDP of a country just by giving some attribute as input for that country.

- **FUTURE SCOPE**

In the future, we can model the system using a vector auto regression algorithm which predictsthe future GDP per capita based upon the GDP per capita in the preceding years.

Here we tried to make research on and implement almost every most possible aspect that will make impact on our proof of concept. In future work we can think of more other sectors that aremaking effect on GDP. After deep and more working on algorithms we can implement new algorithms that will work strongly with data.

- **BIBLIOGRAPHY**

- https://www.kaggle.com
- https://github.com

**APPENDIX:**

Source code:

```
#Importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import seaborn as sns
%matplotlib inline



## Loading the dataset

#Importing the dataset
Development = pd.read_csv("HDI.csv")
#Listing the first five rows of the dataset
Development.head()

## Listing the first 20 rows for better results of visualization
data1= Development.nlargest(20, "HDI")
data1
Development["Country"].unique()



#Data Exploration
#Country
```

```
g = sns.stripplot(x="Country", y='HDI', data=data1, jitter=True)
plt.xticks(rotation=90)


#Data Exploration
#Mean Yearsof Schooling

g = sns.stripplot(x="Mean years of schooling", y="HDI", data=data1, jitter=True)
plt.xticks(rotation=90)

#Data Exploration
#Life Expectancy
g = sns.stripplot(x="Life expectancy", y="HDI", data=data1, jitter=True)
plt.xticks(rotation=90)

#Data Exploration
#Gross national income (GNI) per capita

g = sns.stripplot(x="Gross national income (GNI) per capita", y="HDI", data=data1,
jitter=True)
plt.xticks(rotation=90)

#Data Exploration
#GNI per capita rank minus HDI rank
g = sns.stripplot(x="GNI per capita rank minus HDI rank", y="HDI", data=data1, jitter=True)
plt.xticks(rotation=90)
```

```
#Data visualization

sns.distplot(Development['HDI'])


#Building the correalation matrix

heat = Development.iloc[:,[0,1,2,3,4,5,6,7,67]]
sns.heatmap(heat.corr())
Development.shape


## Label Encoding
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
Development['Country']= le.fit_transform(Development['Country'])

mapping_dict ={}
category_col=["Country"]
for col in category_col:
    le_name_mapping = dict(zip(le.classes_,
                        le.transform(le.classes_)))

    mapping_dict[col]= le_name_mapping
    print(mapping_dict)

Development.head()
```

```
## Selecting the independent and dependent columns

#Independent Variables
X = Development.iloc[:,[2,5,6,7,67]]
X=pd.DataFrame(X)

#Dependant Variable
y = Development.iloc[:,4].values
y=pd.DataFrame(y)

## Checking the null values

#finding the sum of null values in the selected coulumns
X.isnull().sum()

#replacing the null values with the mean
X.fillna(X.mean(),inplace=True)
X.isnull().sum()


## Train and test split

#Train and test split
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

## Model Building

```
#Linear Regression
from sklearn.linear_model import LinearRegression
reg = LinearRegression().fit(x_train, y_train)

#Train and test score
print("Train: ",reg.score(x_train,y_train))
print("Test: ",reg.score(x_test,y_test))

#predicting and printing the result
y_pred=reg.predict(x_test)
print(y_pred)
#calculating the R squared value
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

## Predicting the results

```
x_test

#testing with few values
y_pred=reg.predict([[13,72.0,5.2,3341.0,14.4]])
print(y_pred)

#y_test Values
```

```
y_test

#y_pred values
y_pred

#saving our model into a file
pickle.dump(reg,open('HDI.pkl','wb'))
```