TABLE OF CONTENTS

DECLARATION

CERTIFICATE

ACKNOWLEDGEMENT

ABSTRACT

TABLE OF CONTENTS

INTRODUCTION

- LINEAR REGRESSION

WHY LINEAR REGRESSION IS IMPORTANT

ASSUMPTIONS OF EFFECTIVE LINEAR REGRESSION

IMPLEMENTATION LINEAR REGRESSION WITH SCIKIT- LEARN

- DECISION TREE

HOW DOES DECISION TREE ALGORITHM WORK ?

IMPLEMENTATION

- RANDOM FOREST

WORKING OF RANDOM FOREST ALGORITHM .

IMPORTANT FEATURE OF RANDOM FOREST .

IMPLEMENTATION

CONCLUSION

REFERENCES

ABSTRACT

One of the most pertinent theoretical topics in the field of labour and behavioural economics research is likely the female labour force participation (FLFP). A variety of statistical models have been employed to assess the applicability of explanatory factors. The choice to enter the job market, however, can also be treated as a binary classification problem.

To estimate the female labour force participation, we compare three approaches in this research. Random Forest, Decision Tree, and Linear Regression. The comparison, which was done using data from the World Bank Survey, highlights the benefits and drawbacks of the various sex methodological paradigms and could serve as a fundamental driver for integrating the most effective strategies.

INTRODUCTION

One of the most pertinent theoretical concerns within the purview of studies of both labour and behavioural economics is the labour force participation and its determinants. Furthermore, it can be argued that the decision-making process for labour participation is, at least in part, related to some of the fundamental theoretical frameworks of the field of economics, such as the conventional method of aggregate supply and the more thorough general equilibrium models with micro foundations.

Early labour force participation theories typically focused on how individuals chose to divide their limited free time between work and leisure. The individual's endowment of time is finite in this fundamental framework, thus the final distribution will be determined by the utility-maximizing bundle of salaries (the market value of employment) and free time, which is determined by the relative prices of those two things.

Later, this fundamental framework was expanded to employ the family as the fundamental unit of decision-making for market labour, making interactions and comparative advantages among family members significant for the sake of individual labour decisions. A stable statistical relationship between labour force participation and factors related to the relative cost aforementioned, such as labour market conditions, wages of other family members, education level, fertility rate, and marital status, among others, has been found as an empirical regularity in specialised literature.

Men are more likely than women to participate in the work market in practically every nation in the world. These gender disparities in participation rates have, however, been significantly closing in recent decades. We go down the how and why of these adjustments in this post.

This article's first section gives an overview of the "stylized facts," including information on how women are represented in the informal sector and unpaid caregiving. The second section gives a summary of the major elements that have been fueling the general trends.

Before we move on to the details, here is a preview of the main points:

- Men typically participate in the work market more frequently than women do in most nations.

- Over the past century, women of working age have become more and more involved in the labour force globally.

- The historically rising rate of female labour force participation has slowed or even reversed slightly in several parts of the world in recent years.

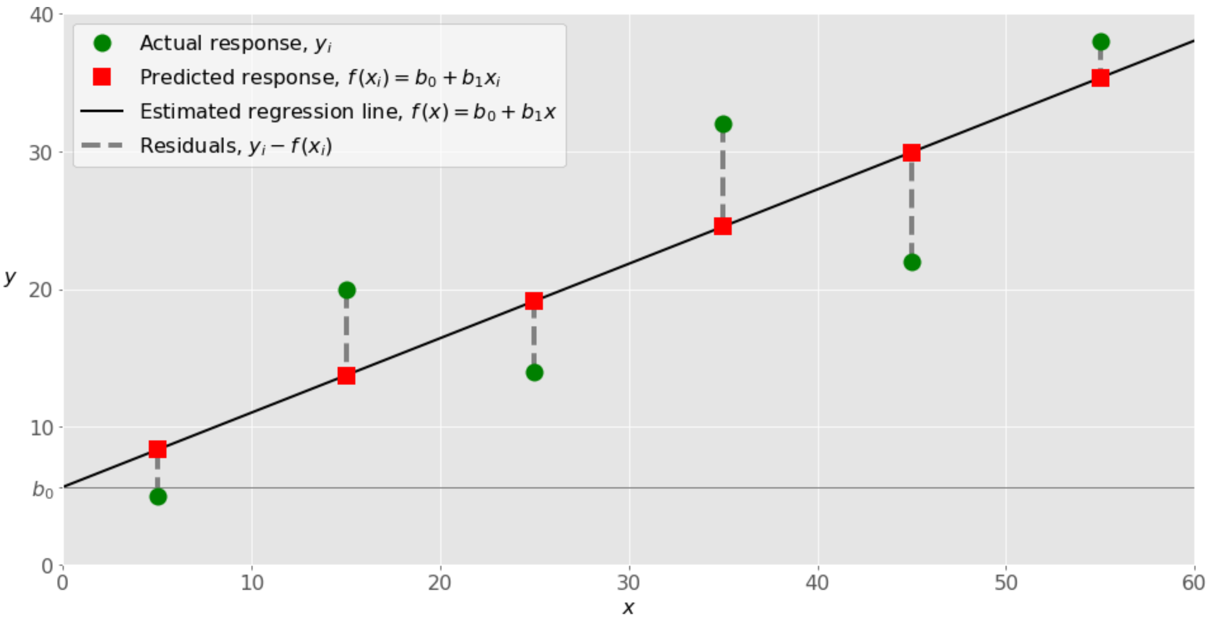
- Worldwide, women devote a significant amount of time to pursuits that aren't traditionally counted as "economic activities."
Therefore, when the timecost of unpaid care work is decreased, shared equally with men, and/or made more compatible with market activity, female involvement in labour markets tends to rise.

LINEAR REGRESSION

A fundamental and widely used form of predictive analysis is linear regression. Regression analysis' main goal is to look at two things: (1) Is it possible to accurately forecast an outcome (dependent) variable using a set of predictor variables? (2) Which individual variables—as shown by the size and sign of the beta estimates—are highly important predictors of the outcome variable, and how do they affect the outcome variable? The link between one dependent variable and one or more independent variables is explained using these regression estimations.

The formula y = c + b*x, where y is the estimated score of the dependent variable, c is a constant, b is the regression coefficient, and x is the score on the independent variable, defines the simplest form of the regression equation with one dependent and one independent variable. There are many names for a regression's dependent variable. It may be called an outcome variable, criterion variable, endogenous variable, or regressand. The independent variables can be called exogenous variables, predictor variables, or regressors.

3 major uses for regression analysis are (1) determining the strength of predictors, (2) forecasting an effect, and (3) trend forecasting.



Why Linear Regression is important ?

The mathematical technique used in linear-regression models is straightforward and can be used to make predictions. Numerous corporate and academic disciplines can benefit from the use of linear regression.
From the biological, behavioural, environmental, and social sciences to business, linear regression is employed widely. Future predictions can now be made scientifically and with high reliability using linear-regression models. The features of linear-regression models are well understood and can be trained extremely quickly since linear regression is a statistical technique that has been around for a very long time.

Assumptions of effective linear regression

Assumptions to be considered for success with linear-regression analysis:

- **For each variable**: Consider the number of valid cases, mean and standard deviation.
- **For each model**: Consider regression coefficients, correlation matrix, part and partial correlations, multiple R, R2, adjusted R2, change in R2, standard error of the estimate, analysis-of-variance table, predicted values and residuals. Also, consider 95-percent-confidence intervals for each regression coefficient, variance-covariance matrix, variance inflation factor, tolerance, Durbin-Watson test, distance measures (Mahalanobis, Cook and leverage values), DfBeta, DfFit, prediction intervals and case-wise diagnostic information.
- **Plots**: Consider scatterplots, partial plots, histograms and normal probability plots.
- **Data**: Dependent and independent variables should be quantitative. Categorical variables, such as religion, major field of study or region of residence, need to be recoded to binary (dummy) variables or other types of contrast variables.
- **Other assumptions**: For each value of the independent variable, the distribution of the dependent variable must be normal. The

variance of the distribution of the dependent variable should be constant for all values of the independent variable. The relationship between the dependent variable and each independent variable should be linear and all observations should be independent.

Python Packages for Linear Regression

A basic scientific Python library called **NumPy** enables a wide range of high-performance operations on both single-dimensional and multidimensional arrays. Numerous mathematical operations are also provided.

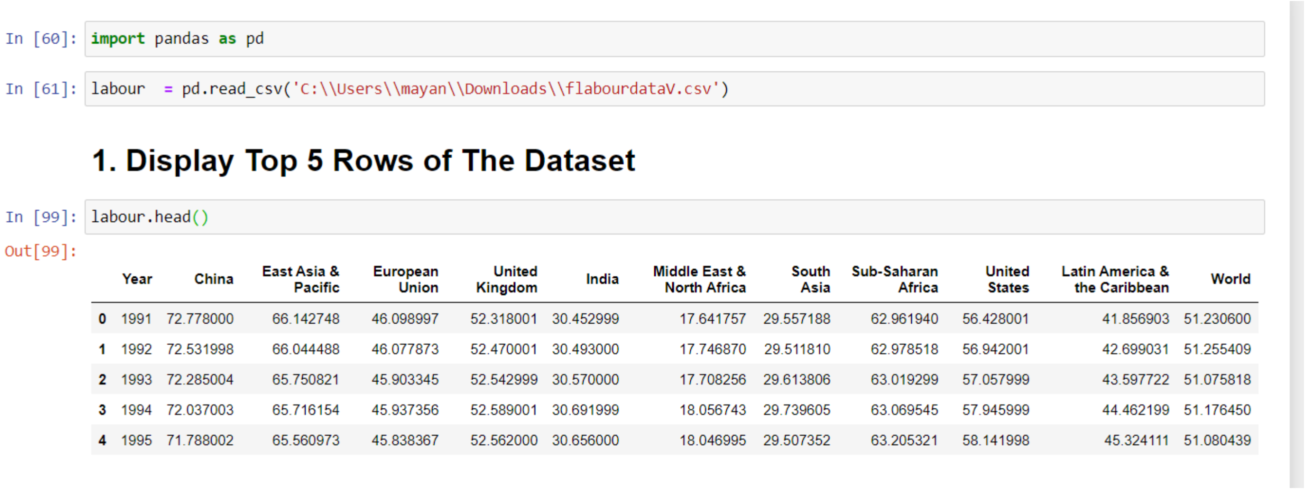
A popular Python machine learning library called **scikit-learn** was created on top of NumPy and a few additional libraries. It offers the tools for data pre-processing, dimensionality reduction, regression implementation, classification, clustering, and more. Scikit-Learn is also open-source, just like NumPy.

Implementation Linear Regression With scikit-learn

We'll start with the simplest case, which is simple linear regression. There are five basic steps when we're implementing linear regression:

1. Import the packages and classes that you need.
2. Provide data to work with, and eventually do appropriate transformations.
3. Create a regression model and fit it with existing data.
4. Check the results of model fitting to know whether the model is satisfactory.
5. Apply the model for predictions.

These steps are more or less general for most of the regression approaches and implementations. Throughout the rest of the tutorial, you'll learn how to do these steps for several different scenarios.



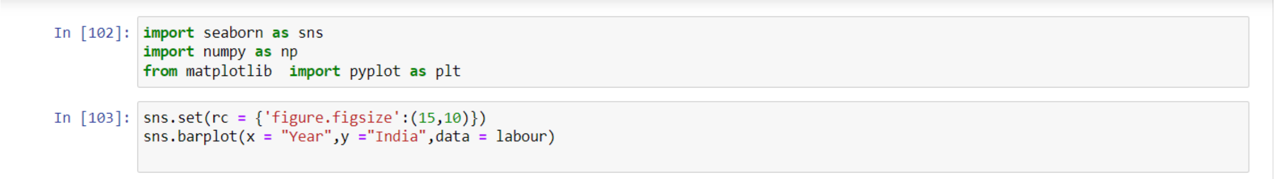
```
In [60]: import pandas as pd

In [61]: labour  = pd.read_csv('C:\\Users\\mayan\\Downloads\\flabourdataV.csv')
```

1. Display Top 5 Rows of The Dataset

```
In [99]: labour.head()
Out[99]:
```

	Year	China	East Asia & Pacific	European Union	United Kingdom	India	Middle East & North Africa	South Asia	Sub-Saharan Africa	United States	Latin America & the Caribbean	World
0	1991	72.778000	66.142748	46.098997	52.318001	30.452999	17.641757	29.557188	62.961940	56.428001	41.856903	51.230600
1	1992	72.531998	66.044488	46.077873	52.470001	30.493000	17.746870	29.511810	62.978518	56.942001	42.699031	51.255409
2	1993	72.285004	65.750821	45.903345	52.542999	30.570000	17.708256	29.613806	63.019299	57.057999	43.597722	51.075818
3	1994	72.037003	65.716154	45.937356	52.589001	30.691999	18.056743	29.739605	63.069545	57.945999	44.462199	51.176450
4	1995	71.788002	65.560973	45.838367	52.562000	30.656000	18.046995	29.507352	63.205321	58.141998	45.324111	51.080439

DATA SET OF FEMALE LABOUR PARTIPATION



```
In [102]: import seaborn as sns
          import numpy as np
          from matplotlib  import pyplot as plt

In [103]: sns.set(rc = {'figure.figsize':(15,10)})
          sns.barplot(x = "Year",y ="India",data = labour)
```

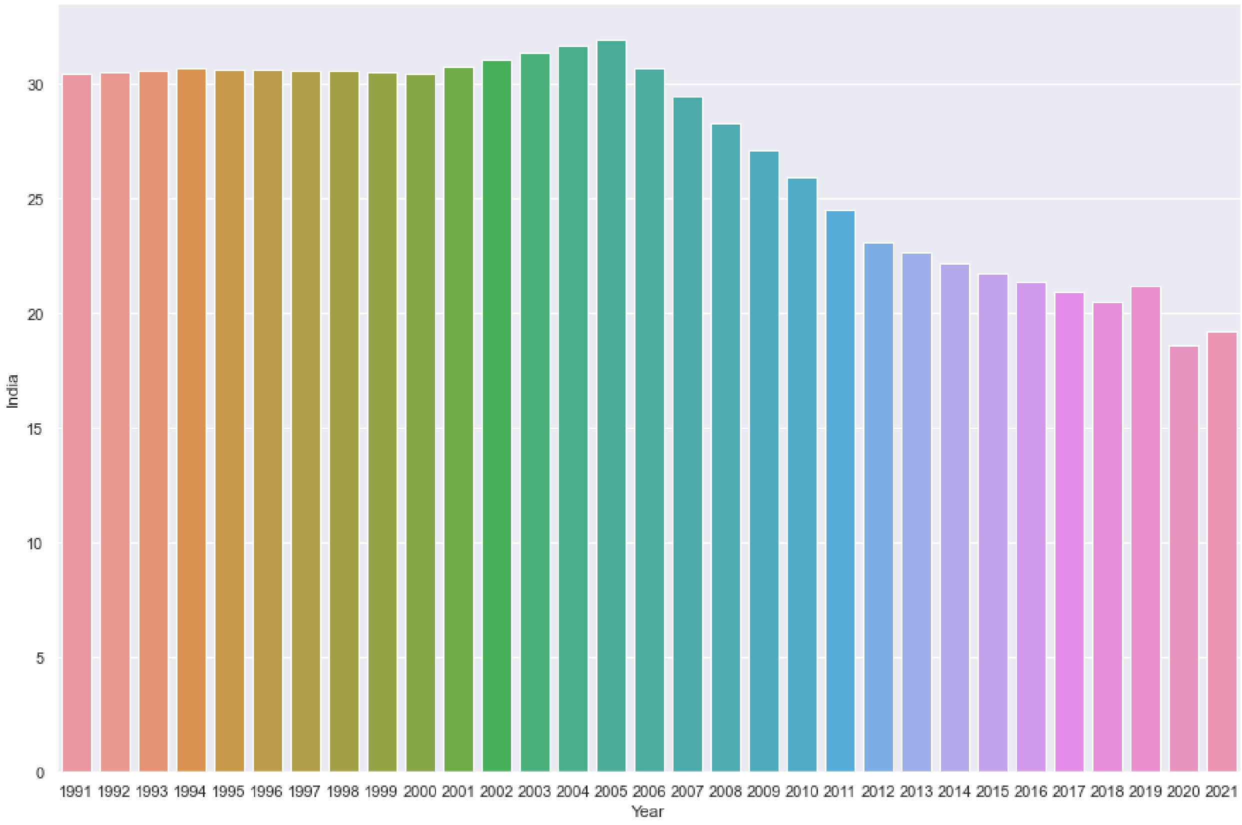


FIG. BAR CHART OF INDIA FEMALE LABOUR PARTICIPATION(1991-2021)

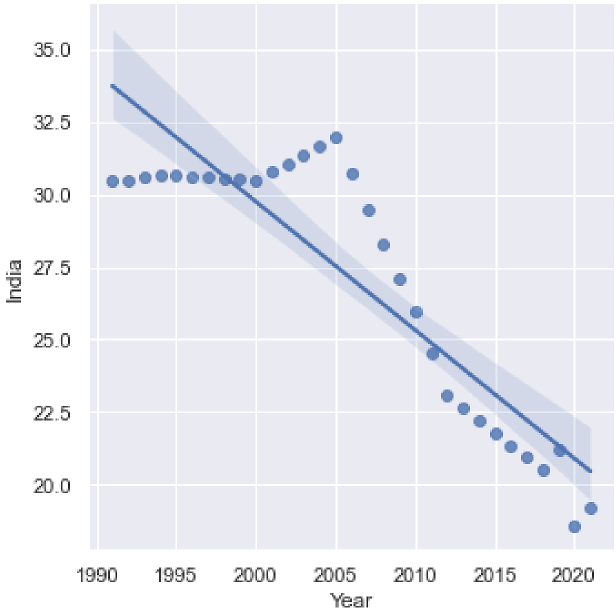
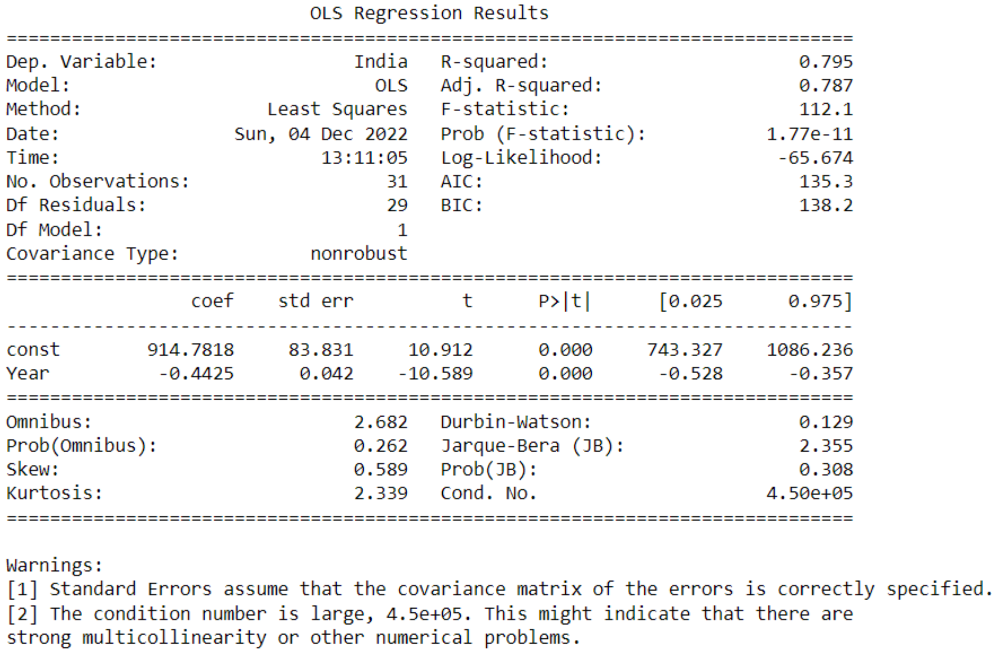


Fig . Linear Regression Plot



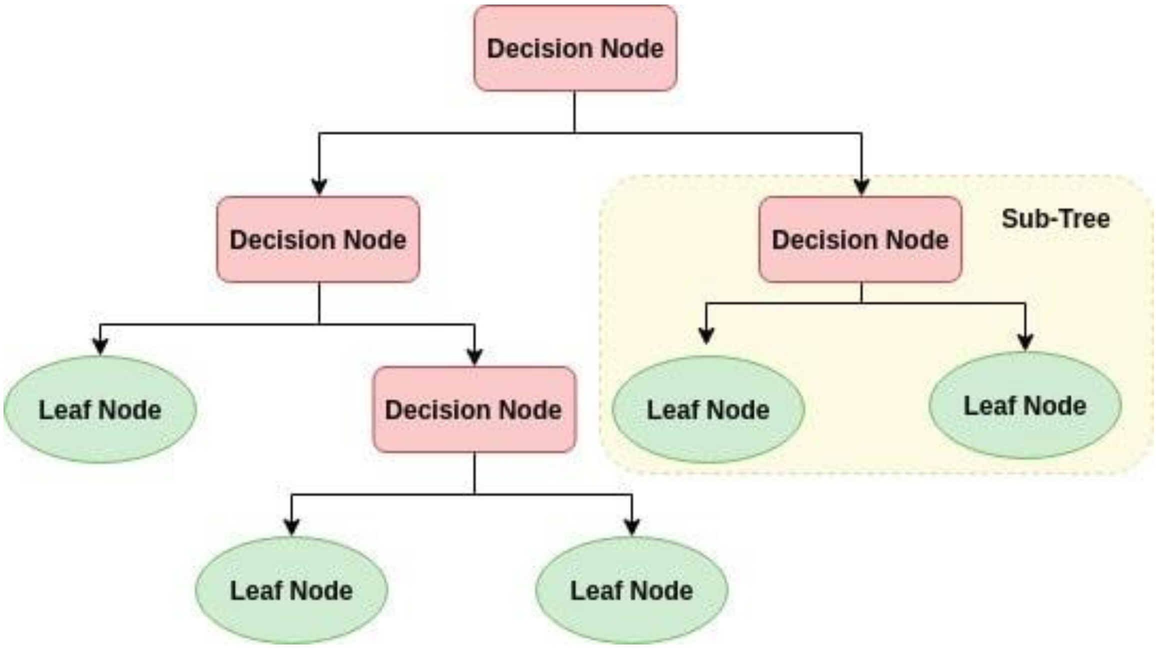
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  India   R-squared:                       0.795
Model:                            OLS   Adj. R-squared:                  0.787
Method:                 Least Squares   F-statistic:                     112.1
Date:                Sun, 04 Dec 2022   Prob (F-statistic):           1.77e-11
Time:                        13:11:05   Log-Likelihood:                -65.674
No. Observations:                  31   AIC:                             135.3
Df Residuals:                      29   BIC:                             138.2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        914.7818     83.831     10.912      0.000     743.327    1086.236
Year          -0.4425      0.042    -10.589      0.000      -0.528      -0.357
==============================================================================
Omnibus:                        2.682   Durbin-Watson:                   0.129
Prob(Omnibus):                  0.262   Jarque-Bera (JB):                2.355
Skew:                           0.589   Prob(JB):                        0.308
Kurtosis:                       2.339   Cond. No.                     4.50e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.5e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Table :- Some Linear Regression Result

DECISION TREE

A decision support tool known as a decision tree employs a tree-like graph or model to represent decisions and all possible outcomes, including utility, resource costs, and chance-event outcomes. View the illustration to get an idea of what it appears to be.

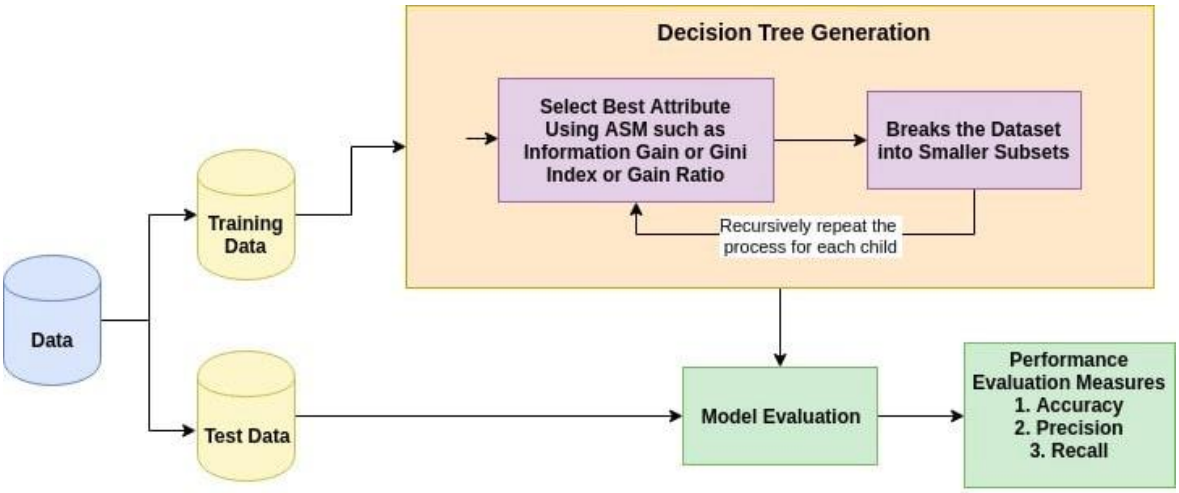
An internal node represents a feature (or property), a branch represents a decision rule, and each leaf node indicates the conclusion in a decision tree, which resembles a flowchart. The root node in a decision tree is the first node from the top. It gains the ability to divide data according to attribute values. Recursive partitioning is the process of repeatedly dividing a tree. This framework, which resembles a flowchart, aids in decision-making. It is a flowchart-like visualisation that perfectly replicates how people think. Decision trees are simple to understand and interpret because of this.



How does the Decision Tree Algorithm Work?

Any decision tree algorithm's fundamental principle is as follows:
1. To divide the records, choose the best attribute using Attribute Selection Measures (ASM).
2. Break the dataset up into smaller subsets and make that attribute a decision node.
3. Recursively repeats this method for each kid to begin growing the tree until one of the conditions is met:

- The identical property value applies to each and every tuple.
- There are no more characteristics left.
- No more occurrences exist.


Decision Tree Generation
Select Best Attribute Using ASM such as Information Gain or Gini Index or Gain Ratio
Breaks the Dataset into Smaller Subsets
Recursively repeat the process for each child
Training Data
Data
Test Data
Model Evaluation
Performance Evaluation Measures
1. Accuracy
2. Precision
3. Recall

Decision Tree Classifier Building in Scikit-learn

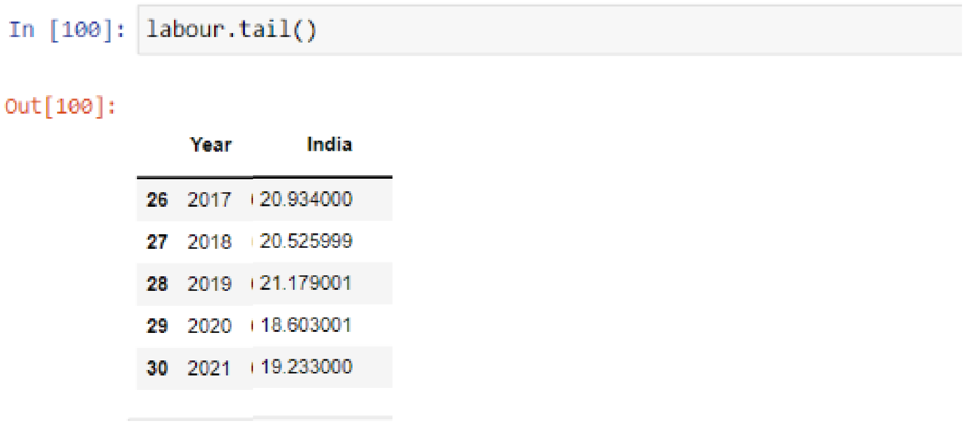
Importing Required Libraries

Let's first load the required libraries.

```
# Load libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split
function
```

```
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
```

Loading Data



```
In [100]: labour.tail()
```

```
Out[100]:
```

	Year	India
26	2017	20.934000
27	2018	20.525999
28	2019	21.179001
29	2020	18.603001
30	2021	19.233000

Splitting Data

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

Let's split the dataset by using function train_test_split(). You need to pass 3 parameters features, target, and test_set size.

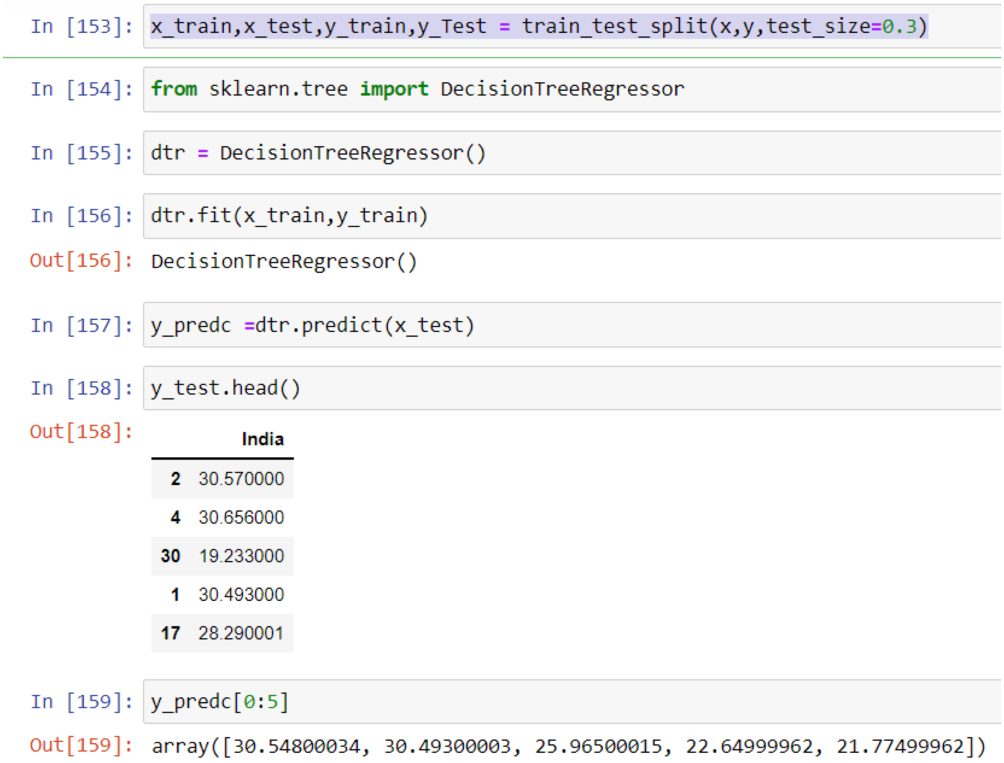
Building Decision Tree Model

Let's create a Decision Tree Model using Scikit-learn.

Evaluating Model

Let's estimate, how accurately the classifier or model can predict the type of cultivars.

Accuracy can be computed by comparing actual test set values and predicted values.



```
In [153]: x_train,x_test,y_train,y_Test = train_test_split(x,y,test_size=0.3)
```

```
In [154]: from sklearn.tree import DecisionTreeRegressor
```

```
In [155]: dtr = DecisionTreeRegressor()
```

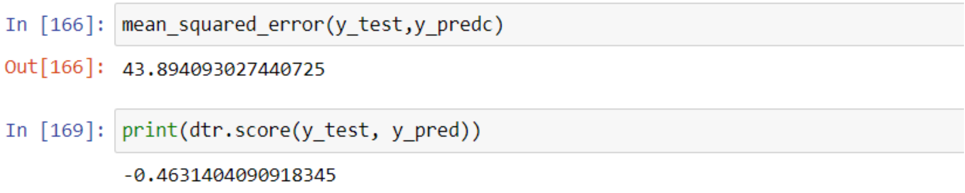
```
In [156]: dtr.fit(x_train,y_train)
Out[156]: DecisionTreeRegressor()
```

```
In [157]: y_predc =dtr.predict(x_test)
```

```
In [158]: y_test.head()
Out[158]:
```

	India
2	30.570000
4	30.656000
30	19.233000
1	30.493000
17	28.290001

```
In [159]: y_predc[0:5]
Out[159]: array([30.54800034, 30.49300003, 25.96500015, 22.64999962, 21.77499962])
```



```
In [166]: mean_squared_error(y_test,y_predc)
Out[166]: 43.894093027440725
```

```
In [169]: print(dtr.score(y_test, y_pred))
          -0.4631404090918345
```

RANDOM FOREST

Random forest and other supervised machine learning algorithms are frequently used in classification and regression problems. It builds decision trees from different samples, using their average for categorization and majority vote for regression.

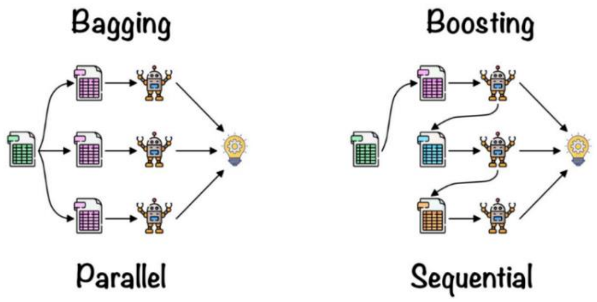
One of the most important features of the Random Forest Algorithm is its capacity to handle data sets containing both continuous variables, as in regression, and categorical variables, as in classification. It produces better results when it comes to classification problems.

Working of Random Forest Algorithm

Before understanding the working of the random forest algorithm in machine learning, we must look into the ensemble technique. ***Ensemble*** simply means combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.

Ensemble uses two types of methods:

1. Bagging: A different training subset is constructed using replacement from sample training data, and the final decision is based on majority voting. Think about Random Forest.
2. Boosting— This method transforms weak learners into strong ones by creating sequential models with the highest level of accuracy. For instance, ADA BOOST and XG BOOST.



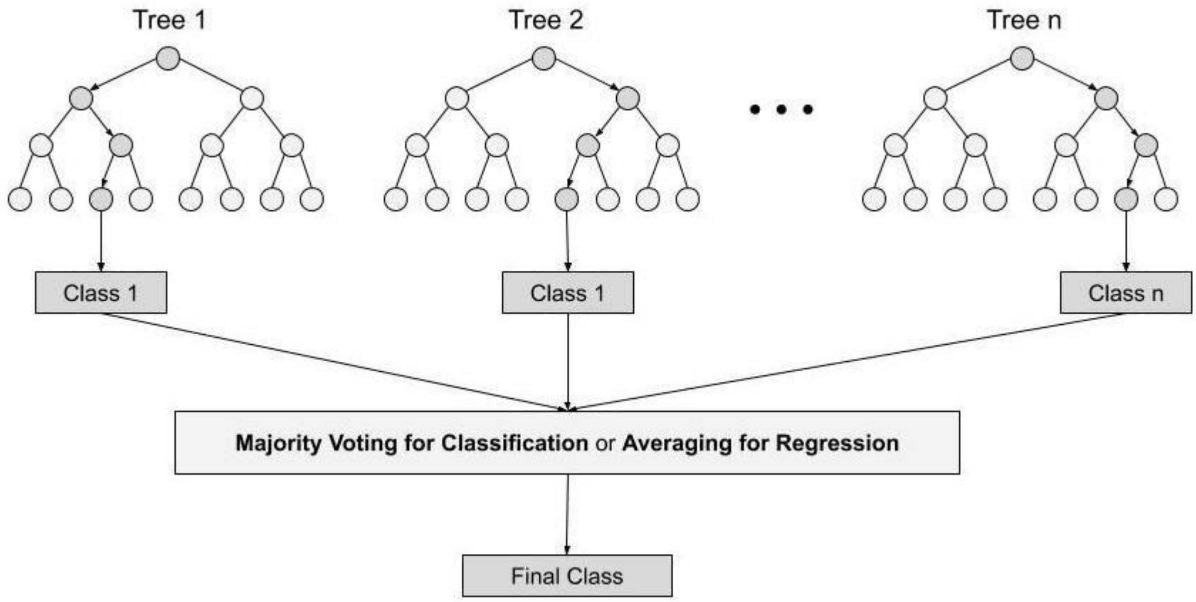
Steps involved in random forest algorithm:

Step 1: In Random forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on ***Majority Voting or Averaging*** for Classification and regression respectively.


Tree 1
Tree 2
Tree n
Class 1
Class 1
Class n
Majority Voting for Classification or Averaging for Regression
Final Class

Important Features of Random Forest

1. Diversity: Since each tree is unique, not all characteristics, variables, or features are taken into account when creating a particular tree.

2. Immune to the dimensionality curse—Because no tree takes into account every feature, the feature space is condensed.

3. Parallelization: Using various data and attributes, each tree is individually generated. This implies that we can create random forests by using the CPU to its fullest extent.

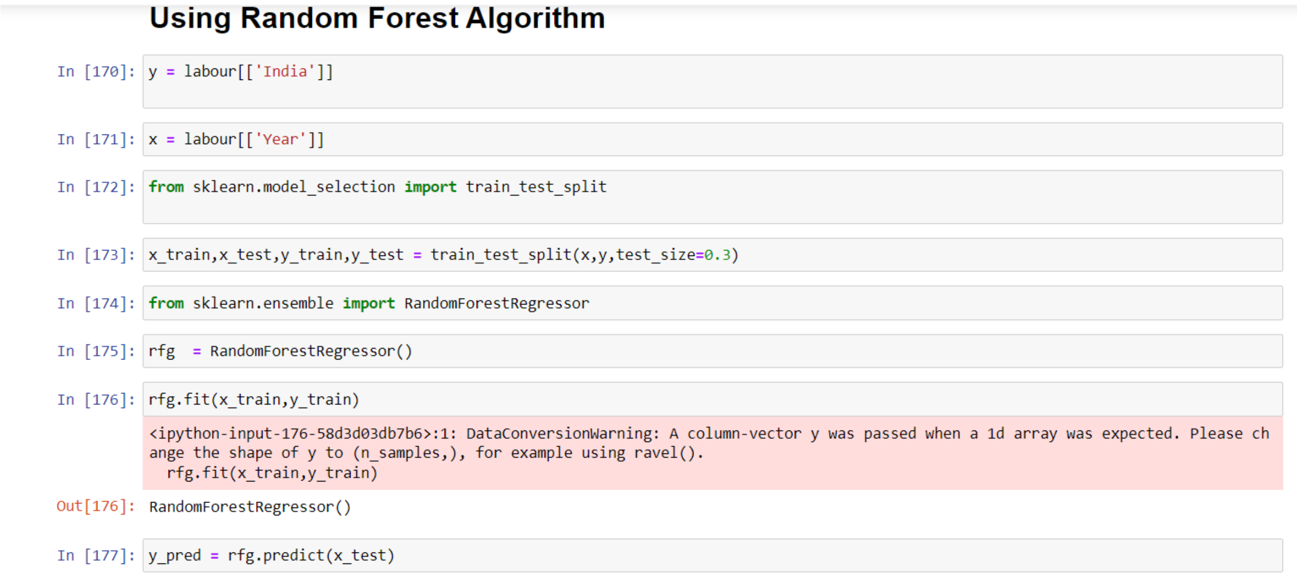
4. Train-Test split: In a random forest, we don't need to divide the data into train and test groups because the decision tree will never view 30% of the data.

5. Stability: Because the outcome is based on majority vote or averaging, there is stability.

Coding in python – Random Forest

Now let’s understand Random Forest with the help of code.

1. Let’s import the libraries.
2. import the dataset.
3. Putting Feature Variable to X and Target variable to y.
4. Train-Test-Split is performed.



Using Random Forest Algorithm

```
In [170]: y = labour[['India']]

In [171]: x = labour[['Year']]

In [172]: from sklearn.model_selection import train_test_split

In [173]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)

In [174]: from sklearn.ensemble import RandomForestRegressor

In [175]: rfg  = RandomForestRegressor()

In [176]: rfg.fit(x_train,y_train)

<ipython-input-176-58d3d03db7b6>:1: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please ch
ange the shape of y to (n_samples,), for example using ravel().
  rfg.fit(x_train,y_train)

Out[176]: RandomForestRegressor()

In [177]: y_pred = rfg.predict(x_test)
In [177]: y_pred = rfg.predict(x_test)
```

```
In [178]: y_test.head(),y_pred[0:5]
```

```
Out[178]: (        India
           21  23.099001
           2   30.570000
           27  20.525999
           26  20.934000
           10  30.771000,
           array([24.05380026, 30.62491944, 21.23738054, 21.45966984, 30.53649929]))
```

```
In [179]: mean_squared_error(y_test,y_pred)
```

```
Out[179]: 0.6475582234215433
```

```
In [188]: from sklearn import metrics
```

```
In [189]: rfg.score(y_test, y_pred)
```

```
Out[189]: -0.8936081407489771
```

CONCLUSION

Algorithms	Mean Square Error
Linear Regression	**24.93**
Decision Tree	**43.89**
Random Forest	**0.64**

In Statistics, Mean Squared Error (MSE) is defined as Mean or Average of the square of the difference between actual and estimated values.

In the Supervised Learning method, the data set contains dependent or target variables along with independent variables. We build models using independent variables and predict dependent or target variables.

The **mean squared error** (MSE) tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It’s called the **mean** squared error as you’re finding the average of a set of errors. The lower the MSE, the better the forecast.

This project has presented an approach to compare classifiers for the Female Labour Force Participation(India) problem, belonging to the statistical and machine learning paradigms. In terms of predictive ability, **the Random Forest performed slightly better than the both algorithms** . Nevertheless, additional analysis could be required to evaluate if statistical hypotheses are fulfilled or not (for example correlation among variables) in order to obtain a better comprehensibility of the estimated coefficient significance. The predictive ability of Random Forest models is better than Linear/Decision Tree models. However if it is very difficult to interpret the information included in support vectors. Moreover, a trial-and-error procedure is required to select the best parameters.

References

1) https://www.researchgate.net/publication/228998196_Estimating_Female_Labor_Force_Participation_through_Statistical_and_Machine_Learning_Methods_A_Comparison

2) https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

3) https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-random-forest-algorithm-for-beginners/

4) https://medium.com/@pranav3nov/decision-tree-classification-5916bba46b1a

5) https://www.ibm.com/in-en/topics/linear-regression

6) https://data.worldbank.org/indicator/SL.TLF.CACT.FE.ZS