PROJECT

ON

<u>Prediction of Bio-oil Yield using Machine Learning</u>

(Using Random Forest Algorithm)

By – Joy Gangopadhyay

B. Tech, National Institute of Technology Warangal

ABSTRACT

Applications of machine learning algorithms (MLAs) to model the Prediction of Bio-oil yield using Machine Learning. A reliable Algorithm, Random Forest Algorithm, was used which could predict Bio-oil yield (R2~0.92) and the dependent variables on which the bio-oil yield is dependent. In this work different Biomass composition analysis (chemical compositions, ultimate analysis and proximate analysis) and pyrolysis conditions (particle size, heating rate and pyrolysis temperature) were successfully used as input to analyze the characteristics of bio-oil by machine learning method.

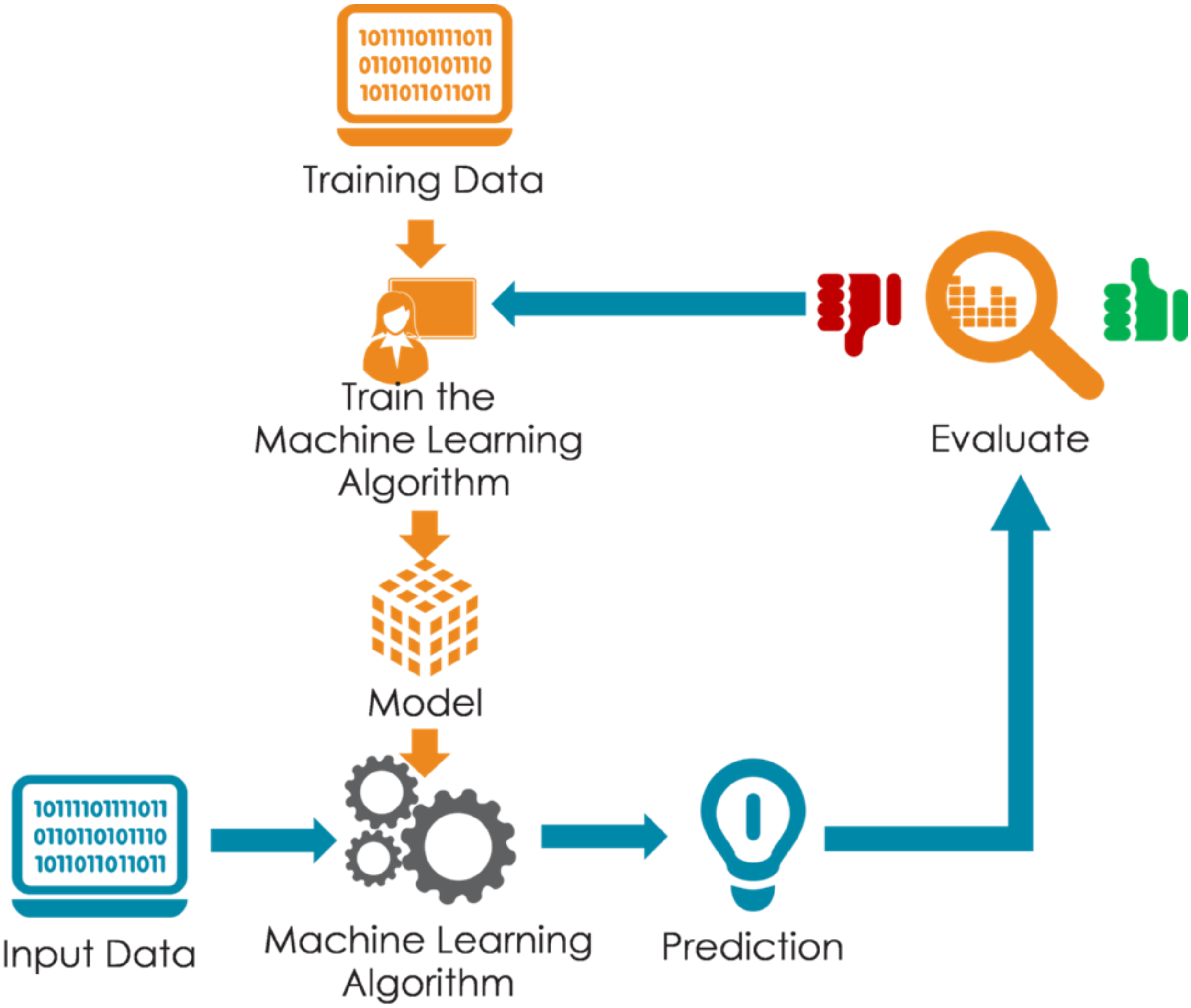
Moreover, root mean squared error analysis and scatter graphs were plotted between all dependent variables and bio-oil yield of all biomasses. This shows the influence of all the variables on the bio-oil yield. The experimental data (282 datasets) from around 30 research paper(s) were carried out to supplement the data.

INTRODUCTON

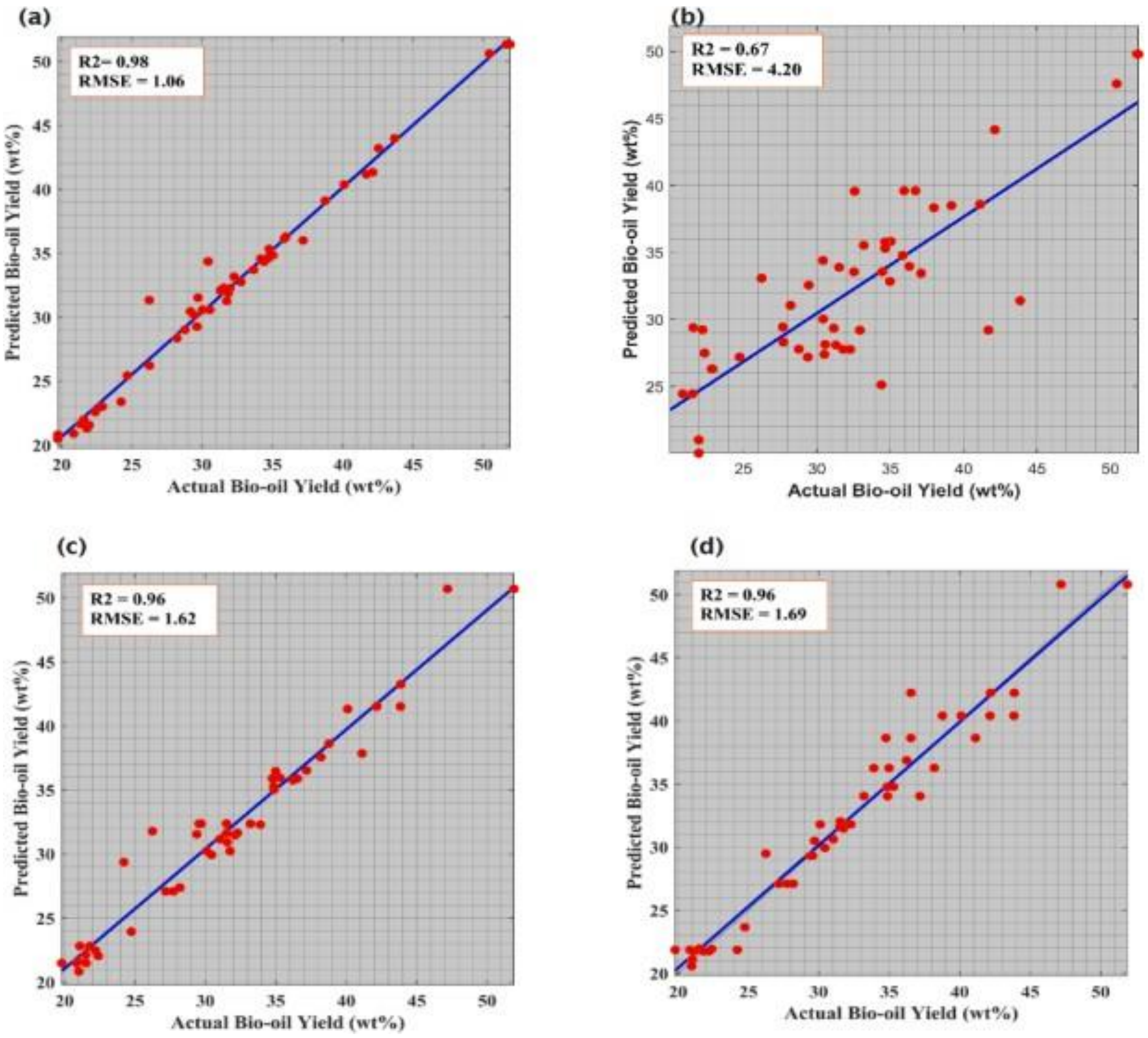
Due to its abundance and renewable nature, biomass energy has received a lot of attention, and the rapid consumption of non-renewable energy has resulted in serious issues with energy security and environmental pollution. Biomass can be converted into biofuels through thermochemical conversion in a cost-effective and efficient manner. The biofuels can then be synthesized into the desired chemicals or used directly. Pyrolysis, a thermal decomposition process under oxygen-free conditions, is considered a potential biomass conversion method due to its relatively simple operation and speed. The main thermochemical conversions of biomass are combustion, gasification, and pyrolysis. Known as bio-oil or pyrolysis-oil, the pyrolysis-oil liquid is typically a dark brown, viscous liquid made up of 350 highly oxygenated compounds. A definitive investigation is utilized to decide the decent carbon, hydrogen, oxygen and nitrogen contents while the general examination is utilized to decide the proper dampness, unpredictable matter and debris items in biomass. The temperature of the pyrolysis, the rate of heating, and the size of the biomass particles all have an impact on the pyrolysis process.

However, the conventional methods for determining the correlation between the bio-oil yield and its influencing factors (biomass composition and pyrolysis conditions) necessitate extensive, time-consuming, and costly experimentation. Consequently, it is essential to use machine learning to effectively analyze the cumulative impact of feedstock composition and pyrolysis process conditions on the behavior of biomass pyrolysis on the

Schematic Representation of a typical Machine Learning Structure:



Comparative analysis of Machine Learning models for prediction of bio-oil yield-



The above figure shows Comparison of experimental and predicted models output data based on proximate analysis and pyrolysis conditions: (a) Random Forest (b) Multi Linear Regression (c) Support Vector Machine (d) Decision Tree.

It is evident from above that the Random Forest algorithm has higherprediction accuracy than others.

Steps-

1) Firstly, in order to improve the accuracy of our bio-oil yield prediction, conducted a thorough comparative analysis of various machine learning (ML) models. We evaluated their RMSEand R2 values to determine the most promising model.

2) Based on the analysis, we ultimately selected the Random Forest Algorithm for its high R^2 value and accuracy.

3) Then searched through numerous research papers to acquire a large dataset consisting of 130-150 different biomass samples, along with their respective bio-oil yield data, Chemical Compositions, Ultimate and Proximate analysis data, and pyrolysis conditions.

4) Next, sorted the data into two parts: the training dataset (80%)and the testing dataset (20%).

5) Finally, tested the model and fine-tuned hyperparameters suchas n_estimators and Random state to further improve the R^2 valueof the model.

6) By following this rigorous methodology, able to build ahighly accurate model for predicting bio-oil yield based on Ultimate and Proximate analysis of the biomass samples.

Random Forest Algorithm

An ensemble supervised learning algorithm is Random Forest. The "backwoods" it fabricates is a group of choice trees, normally prepared with the packing technique. The general premise of the bagging method is that using multiple learning models together improves the end result. In order to produce a prediction that is both more reliable and accurate, Random Forest takes multiple decision trees and combines them.

Ensemble Machine Learning

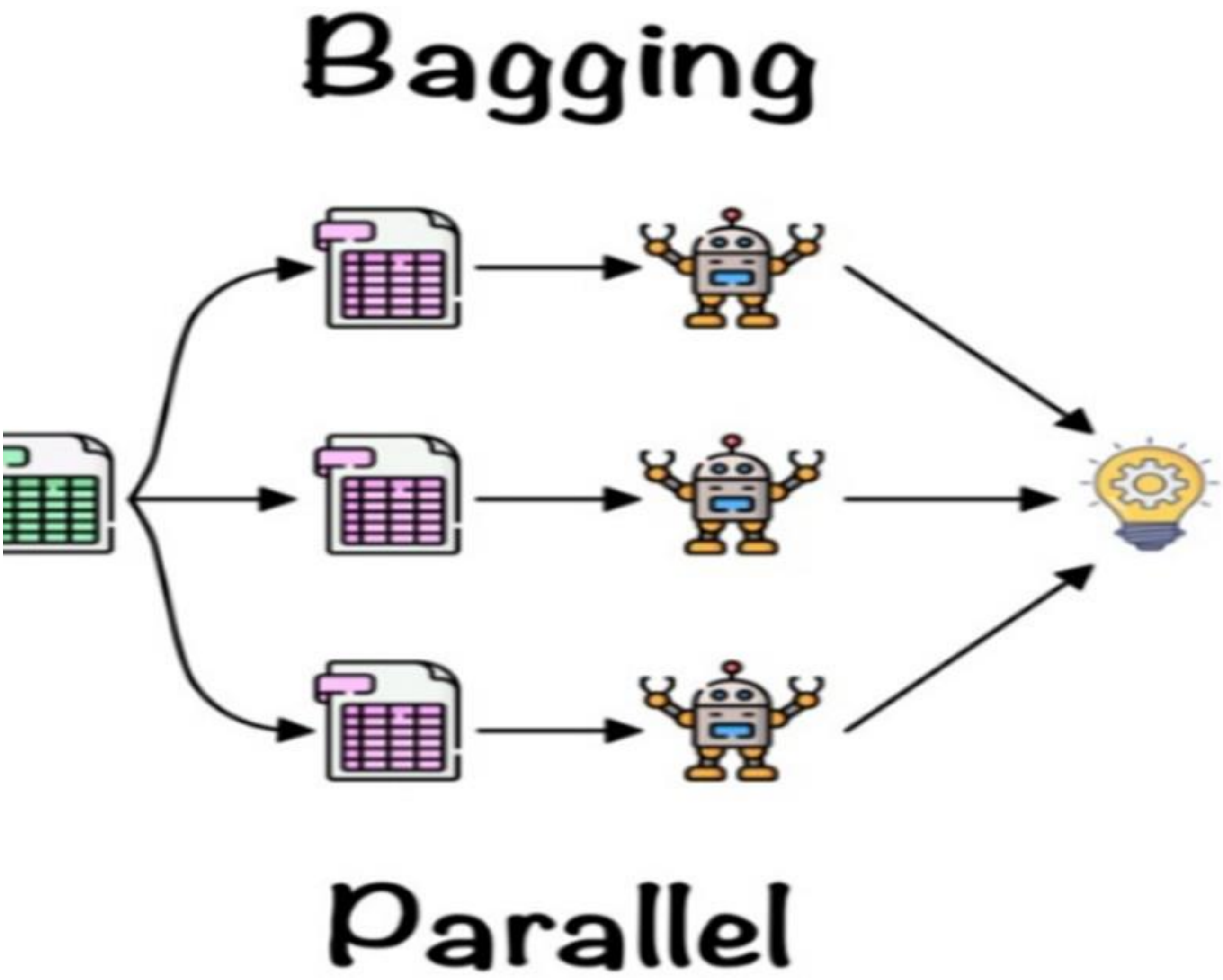
Machine learning ensemble refers to a collection of components viewed collectively rather than separately. To solve the problem, an ensemble method creates multiple models and combines them.

Bagging is the ensemble method we used here.

Bagging

Bagging uses replacement to create a new training subset from sample training data, and the final result is determined by majority vote.
Random Forest employs the ensemble technique known as bagging, which is also referred to as bootstrap aggregation. Thus each model is produced from the examples (Bootstrap Tests) furnished by the First Information with substitution known as line examining. Bootstrap is the process of replacing row sampling steps. The last result depends on a larger part casting a ballot subsequent to joining the consequences, everything being equal. This step which includes joining every one of the outcomes and producing yield in view of larger part casting a ballot is known as collection.

Schematic representation of Bagging process



Working mechanism of Random Forest Algorithm:

Step 1-Random subset selection: The algorithm selects a random subset of the training data and features for each decision tree in the forest.

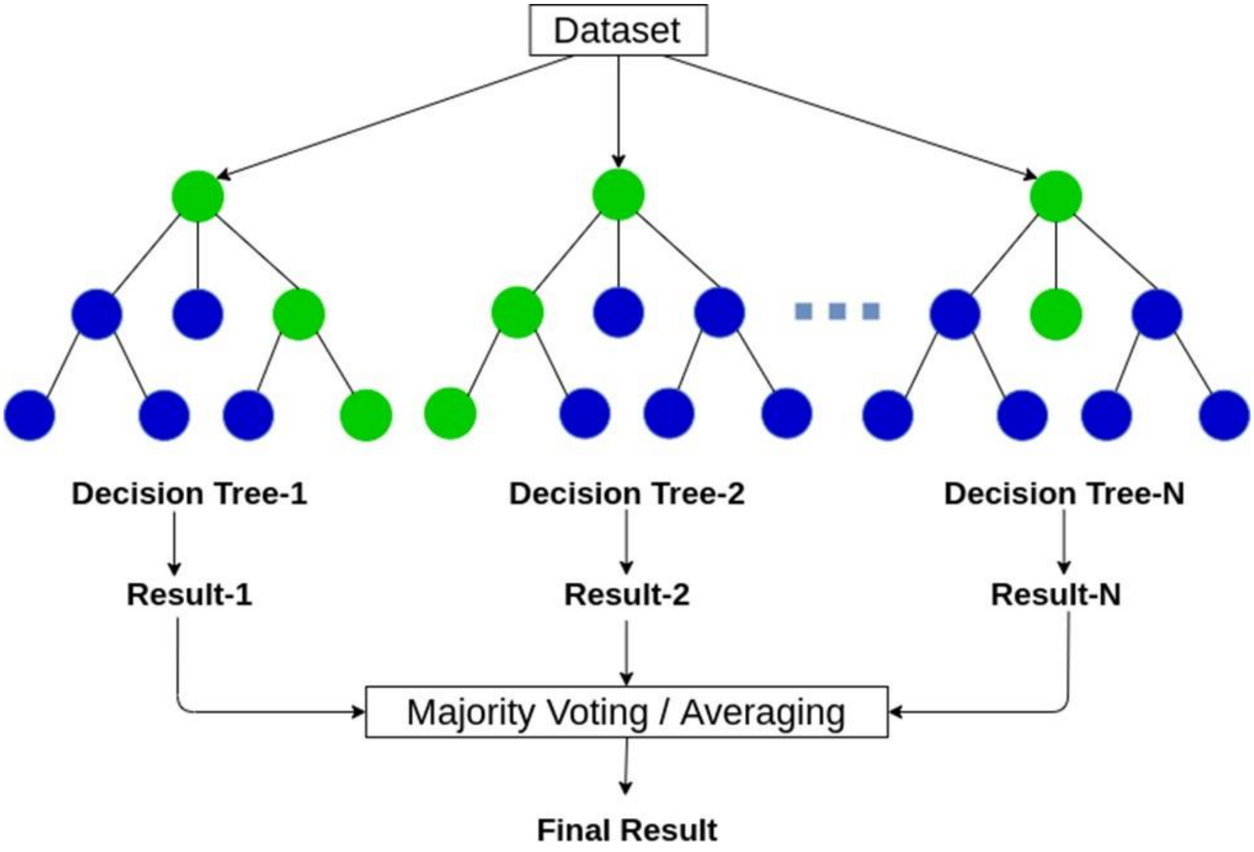
Step 2-Decision tree creation: A decision tree is created for each subset of data and features. Each decision tree is created by recursively splitting the data based on the selected features until a stopping criterion is met.

Step 3-Voting: Each decision tree in the forest independently predicts the outcome, and the results are tallied by a voting mechanism. The majority prediction is selected as the final prediction.

Step 4-Bias reduction: The random selection of subsets of data and features helps reduce bias and overfitting, which can result in a more accurate and robust model.

Step 5-Tuning: Hyperparameters, such as the number of trees in the forest and the maximum depth of each tree, can be tuned to optimize model performance.

Result: The value of R^2 is displayed after processing above steps which gives out the accuracy of the model.



RMSE Value:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(Y_i^{exp} - Y_i\right)^2}$$

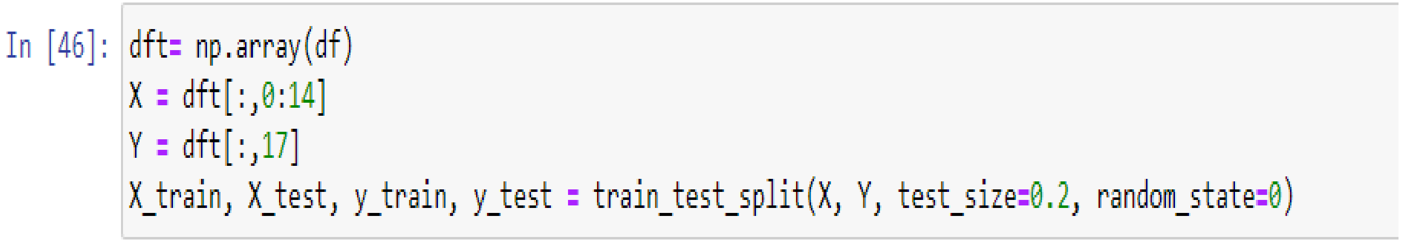
R^2 Value:

$$R2 = 1 - \frac{\sum_{i=1}^{n}\left(Y_i^{exp} - Y_i\right)^2}{\sum_{i}^{n}\left(Y_i^{exp} - Y_{avg}^{exp}\right)^2}$$

The above figure shows theoretical formulae of both R^2 and RMSE.

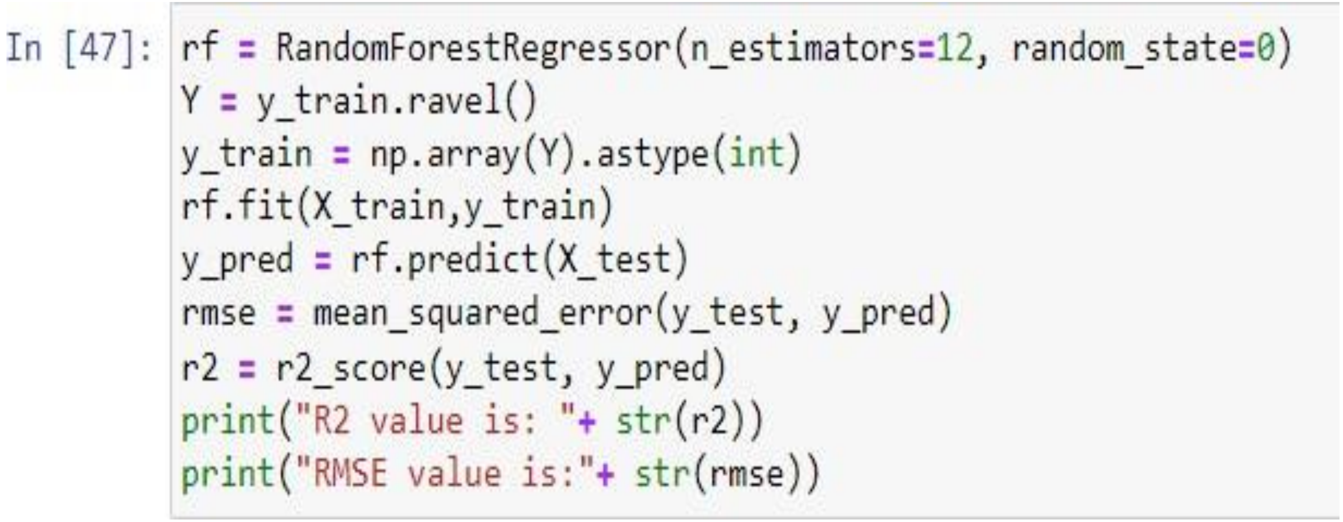
Code:

# Splitting the data into training (80% data) and testing(20% data)



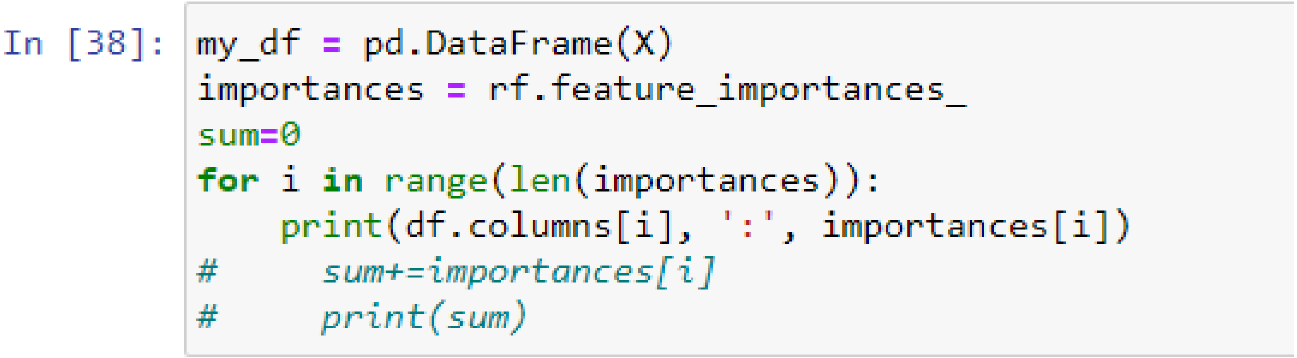
```
In [46]: dft= np.array(df)
         X = dft[:,0:14]
         Y = dft[:,17]
         X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

# Using Random Forest Algorithm to process the data intoa model.



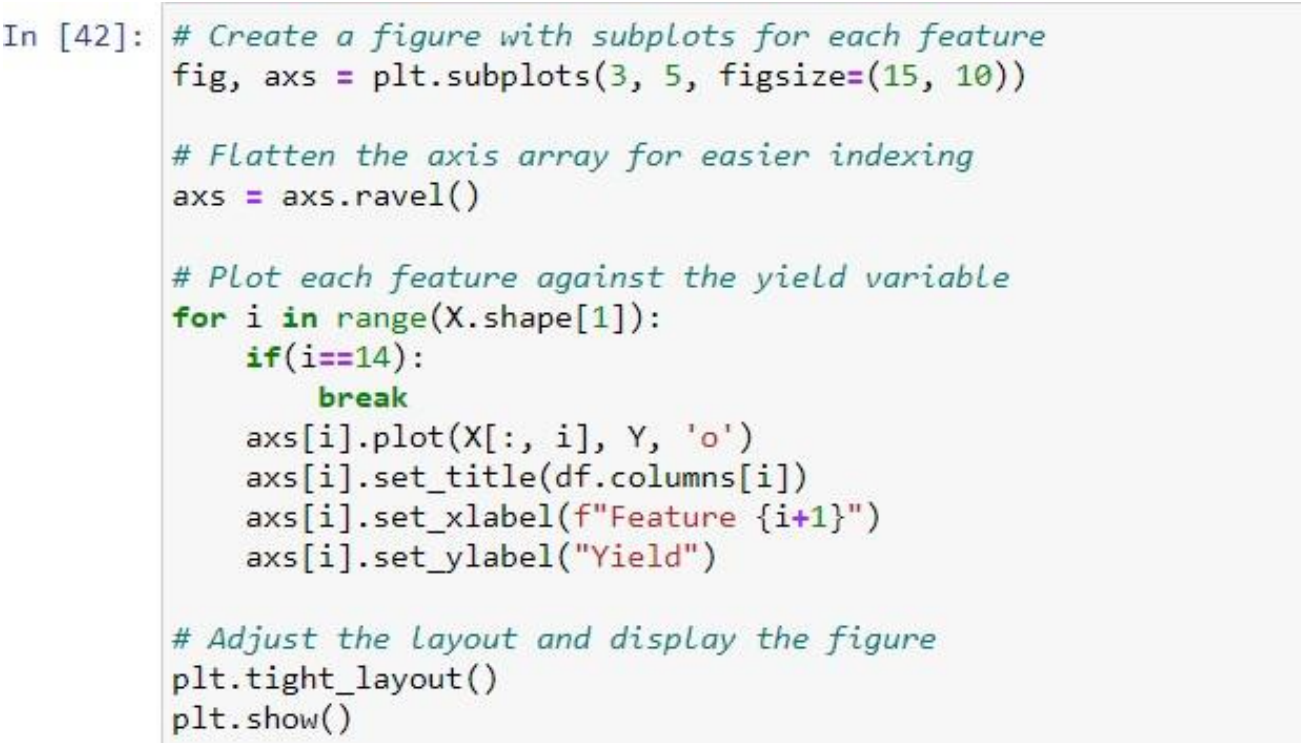
```
In [47]: rf = RandomForestRegressor(n_estimators=12, random_state=0)
         Y = y_train.ravel()
         y_train = np.array(Y).astype(int)
         rf.fit(X_train,y_train)
         y_pred = rf.predict(X_test)
         rmse = mean_squared_error(y_test, y_pred)
         r2 = r2_score(y_test, y_pred)
         print("R2 value is: "+ str(r2))
         print("RMSE value is:"+ str(rmse))
```

# Evaluating the dependent variable.



```
In [38]: my_df = pd.DataFrame(X)
         importances = rf.feature_importances_
         sum=0
         for i in range(len(importances)):
             print(df.columns[i], ':', importances[i])
         #     sum+=importances[i]
         #     print(sum)
```

# Plotting the dependent variable on graphs (dependent variable vs bio-oil yield).



```
In [42]: # Create a figure with subplots for each feature
         fig, axs = plt.subplots(3, 5, figsize=(15, 10))

         # Flatten the axis array for easier indexing
         axs = axs.ravel()

         # Plot each feature against the yield variable
         for i in range(X.shape[1]):
             if(i==14):
                 break
             axs[i].plot(X[:, i], Y, 'o')
             axs[i].set_title(df.columns[i])
             axs[i].set_xlabel(f"Feature {i+1}")
             axs[i].set_ylabel("Yield")

         # Adjust the layout and display the figure
         plt.tight_layout()
         plt.show()
```

OBJECTIVES

1) It involves using a machine learning model to accurately predict the bio-oil yield of any biomass that undergoes pyrolysis.

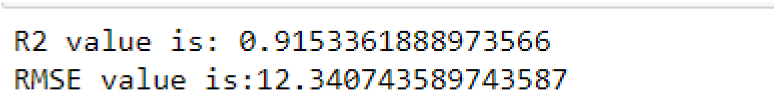
2) To investigate the impact of various pyrolysis parameters, such as temperature, heating rate, and biomass feedstock type, on the bio-oil yield, and identify the most important factors affecting the yield.

3) To compare the performance of the developed machine learning model with other traditional statistical models used for bio-oil yield prediction and determine the superiority of the proposed model.

4) To provide insights into the relationship between pyrolysis parameters, bio-oil yield, and the machine learning model's predictions, and suggest possible optimization strategies to maximize the bio-oil yield

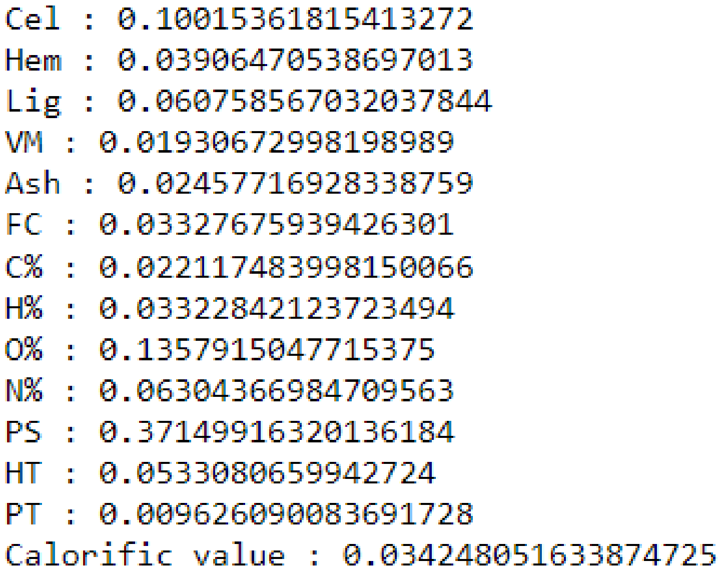
RESULTS AND DISCUSSION

The R^2 and RMSE value are:



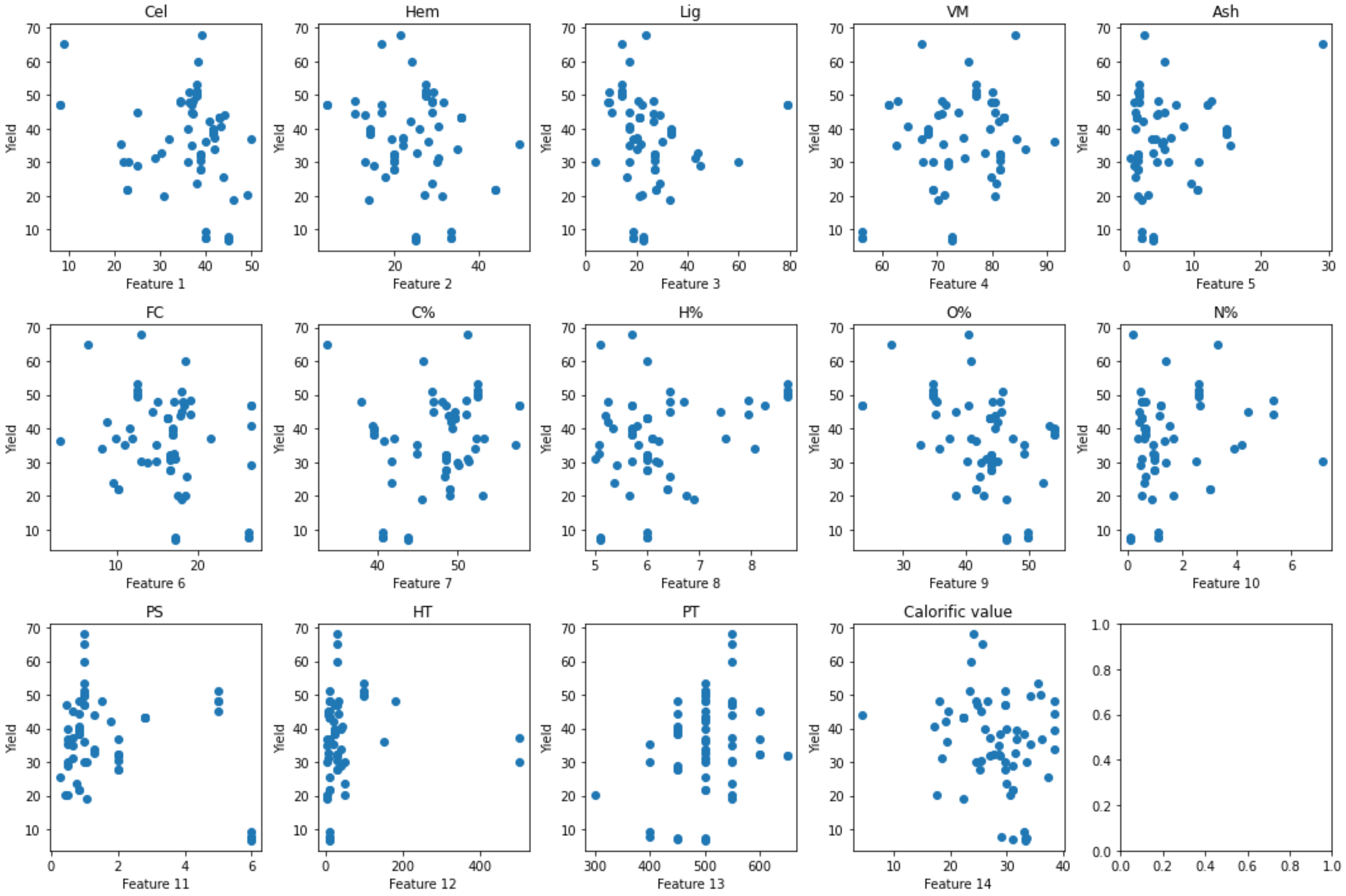
```
R2 value is: 0.9153361888973566
RMSE value is:12.340743589743587
```

Dependency of various parameters affecting the bio-oil yield:



```
Cel : 0.10015361815413272
Hem : 0.03906470538697013
Lig : 0.060758567032037844
VM : 0.01930672998198989
Ash : 0.02457716928338759
FC : 0.03327675939426301
C% : 0.022117483998150066
H% : 0.03322842123723494
O% : 0.1357915047715375
N% : 0.06304366984709563
PS : 0.37149916320136184
HT : 0.0533080659942724
PT : 0.009626090083691728
Calorific value : 0.034248051633874725
```

Various Graphs depicting the correlation between individual parameters and bio-oil yield:



➔The charts above illustrate the relationship between the **14 input parameters** on the x-axis and the resulting bio-oil yield on the y-axis. Through these scatter plots, we can visually analyze the data distribution and identify the optimal range of individual parameters that have the greatest impact on the bio-oil yield.

Results:

- The Yield of bio-oil highly depends majorly on C%, O%, particle size, cellulose content and the heating rate.

- Once the temperature surpasses a certain threshold point that typically corresponds to the maximum bio-oil yield, any further increase in temperature results in a decline of the bio-oil yield.

- With increasing Particle Size greater than 5 mm , the bio-oil yield decreases. The smaller the particle size (<0.5 mm) the higher the bio-oil yield will be.

- From the graph it is evident that high Yield is obtained in the case where the Pyrolysis Temperature is in the range of 450-600 °C.

- The Less the ash content the more the bio-oil yield is. The presence of ash can lead to the formation of char and/or slag, which can decrease the amount of bio-oil.

- During the pyrolysis process, the biomass is heated to a high temperature in the absence of oxygen, and the volatile compounds in the biomass evaporate and are collected as bio-oil. Therefore, biomass with higher volatile matter content can result in higher bio-oil yields.

CONCLUSION:

We have developed a proficient machine learning algorithm that can predict the bio-oil yield (output) using **14 distinct input parameters**. Through this, we have determined the correlation and dependency in a graphical method of each of these 14 parameters with the yield.

Algorithm used: Random Forest.

The Random Forest method depicted that Ultimate Analysis and particle size gave a good amount of contribution towards the bio-oil yield. External factors like heating rate and pyrolysis temperature also affected the Bio-Oil yield. Around **280 datasets** consisting of around **130 biomasses** were used in the form of an Excel File. We can conclude that this algorithm would work properly for predicting bio-oil yield of various biomasses accurately to **~92%(R^2)**. Thus, this can save lots of costs and time and pave the way for future experimental research on only those sets of selected biomasses with