iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XII **Month of publication:** December 2021

DOI: https://doi.org/10.22214/ijraset.2021.39576

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

Machine Learning Based Heat Transfer Optimization of Nano-fluid flow in a Helically Coiled Pipe

Awnish Kumar[1], Dr. Devesh Kumar[2]
[1]*M.Tech, Energy Technology and Management, MMM University of Technology, Gorakhpur*
[2]*Assistant Professor, Department of Mechanical Engineering, MMM University of Technology, Gorakhpur*

Abstract: ***Machine Learning algorithms are widely used in various fields such as energy sectors, manufacturing sectors and aerospace sectors. These algorithms are used mainly in predictive and optimization purpose. The present study deals with the application of two machine learning algorithms i.e. Random Forest algorithm and Support Vector Machine Algorithm to predict the heat transfer efficiency of a flowing nano-fluid in a helically coiled pipe.***
Keywords: Machine Learning; Optimization; Nano-fluid; Heat Transfer

I. INTRODUCTION

Machine learning is a branch of artificial intelligence that is defined as a machine's ability to mimic intelligent human behavior. Artificial intelligence systems are utilized to complete complex jobs in a similar manner to how humans solve problems [1-4].
According to Boris Katz, a primary research scientist and head of the CSAIL's InfoLab Group, the purpose of AI is to construct computer models that demonstrate "intelligent behaviors" similar to humans. This refers to machines that can detect a visual picture, comprehend a natural-language text, or perform a physical activity.
Machine learning is divided into three categories:
Machine learning models that are **supervised** are trained with labeled data sets, which allows the models to learn and improve over time. For example, an algorithm would be trained with images of dogs and other objects, all of which would be classified by humans, and the machine would learn how to recognize images of dogs on its own. The most popular sort of machine learning today is supervised machine learning.
Unsupervised machine learning is a type of machine learning in which a program searches for patterns in unlabeled data. Unsupervised machine learning can uncover patterns or trends that individuals aren't aware of. An unsupervised machine learning software, for example, may shift through online sales data and detect distinct types of customers making purchases.
Reinforcement machine learning uses an incentive system to train machines to do the best action through trial and error. Reinforcement learning can be used to teach models to play games or autonomous vehicles to drive by informing the machine when it has made the correct judgments, allowing it to learn over time what actions to do.
Figure 1 shows the graphical map of the classification of Machine Learning algorithms.

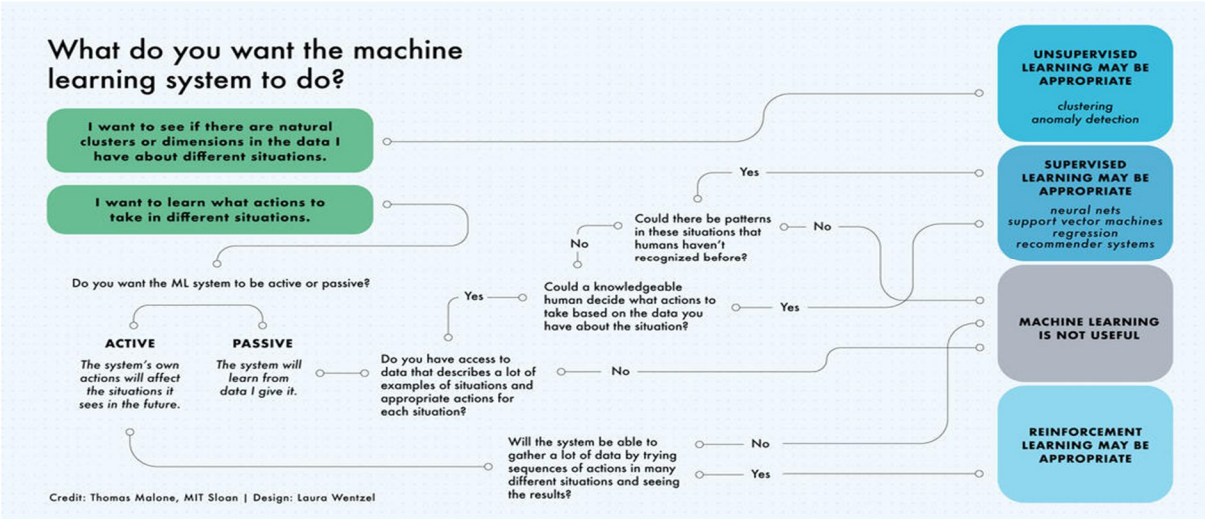


Figure 1: Machine Learning definition and its classification

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

Machine learning and artificial intelligence (AI) are two of the most used commercial terms these days. As a result, businesses across industries are searching for methods to incorporate them into their fundamental processes in order to optimize and automate them [5-10]. And the energy sector is no different!

In reality, the power of machine learning has benefited renewable energy firms (wind, solar, hydro, nuclear) tremendously throughout the years. They were able to reduce their costs, improve their forecasting, and raise the rate of return on their portfolio. And this upward tendency is only likely to accelerate. If your company is in the energy sector – or consumes a lot of electricity – machine learning and AI may be able to help you improve your business performance.

Grid management is one of the most intriguing applications of AI in the energy sector. Customers receive electricity through a sophisticated network (also known as the power grid). The power grid is complicated by the fact that electricity generation and consumption must always be in sync. Otherwise, problems such as blackouts and system failures could occur.

Although there are several methods for storing energy, the most prevalent is pumped hydroelectric storage, which is an ancient but still effective approach. It works by pumping water to a set elevation and then allowing it to fall into turbines to be harnessed again.

It's difficult to forecast the grid's electricity generation capability when dealing with renewable energy. After all, it is influenced by a variety of things, including sunlight and wind.

Predictive maintenance can be implemented using machine learning techniques. In essence, sensors are installed on electricity lines, machinery, and stations to capture operational time series data (data accompanied by a timestamp).

Machine learning algorithms can then forecast if a component will fail in a certain length of time (or n-steps). It can also predict how long a piece of machinery will last or when the next failure will occur. These algorithms' main goal is to accurately forecast machine failure, eliminate blackouts or downtimes, and optimize maintenance operations and frequency, lowering maintenance costs.

Governments and electric corporations aren't the only ones focusing on renewable energy sources. Indeed, several organizations, such as Google and Microsoft, have attempted to reduce their overall energy use in order to have a positive influence on the environment and their bottom lines.

Google is well-known for its gigantic data centers that have been set up all around the world. These data centers generate a lot of heat, which necessitates a lot of electricity to cool down.

DeepMind AI deployed machine learning techniques to cut energy cooling on its Google data centers by 40% to address the issue. It not only helped them save money on their power costs, but it also helped to reduce overall emissions, lowering the carbon tax they would have had to pay otherwise.

For methanol–MOF couples in AHPs/ACs, a computational screening of 6013 computation-ready experimental metal–organic frameworks (CoRE-MOFs) is undertaken, followed by a directional screening of 137 953 hypothetical MOFs (hMOFs) based on the range of optimal CoRE-MOF descriptors. Then, based on the equipment cost (Cequipment), cycle cost (Ccycle), and material cost (Cmaterial), a techno-economic analysis (Ctotal) of AHPs/ACs for each MOF was performed (CMOF). The random forest was found to have the best prediction effect among six machine learning (ML) algorithms, with little effect from the metal type of MOFs, however the heat of adsorption and MOF density were two key characteristics in determining Ctotal. For each application, 12 low-cost CoRE-MOFs and hMOFs were discovered, with average costs in state 2 of under 1 USD kJ1 in AHPs/ACs. Finally, it was expected that a variation of the well-known MOF (Cu3BTC2) would have greater techno-economy, which was proven by a parallel methanol adsorption experiment [11].

For sensible heat thermal energy storage, molten salt with a high specific heat is required. Molten-salt nanofluid is being studied as a possible solution in current scientific studies. The relationship between system factors introduced during nanofluid preparation and specific heat augmentation, on the other hand, is not well understood. We wondered if there was a common pattern/clusters in the nanofluid samples reported in previous studies because finding explicit relationships is difficult due to complicated chemical interactions between molten salt and nanoparticles. Unsupervised machine learning approaches such as Hierarchical cluster analysis (HCA) and Principal component analysis are used to investigate data-driven relationships between samples (PCA). Concentration, temperature, density ratio, and nanoparticle size are the descending order of contribution of the system parameters in the specific heat enhancement percent, according to three main components that captured 81.3 percent variance of the total dataset [12].

The effect of geometrical and operational parameters on nanofluid heat transfer in a microchannel heat sink (MCHS) with two synthetic jets is investigated using computational fluid dynamics (CFD) and machine learning algorithms (MLAs) (SJs). Insert kinds (dimple/protrusion), insert arrangements (inline/staggered), insert sizes, and jet phase actuation (in-phase/180° out-of-phase actuation) are all studied using parametric CFD.

The k-nearest neighbor (kNN), random forest (RF), Gaussian process regression (GPR), and multi-layer perceptron are four prominent regression models used to reduce processing costs (MLP). By adding 5% alumina particles and inline protrusions (D = 0.26 mm) to the MCHS with inactive jets, the heat transfer coefficient (HTC) is increased by 104.8 percent. The heat transmission rate and pressure drop ratio are likewise higher in staggered setups [13].

In the field of renewable and sustainable energy systems, nanofluids are gaining traction in research and development. A tiny amount of high thermal conductivity solid nanoparticles added to a base fluid could improve its thermophysical characteristics and increase heat transfer. Nonlinear effects on the thermodynamics, heat transfer, fluid flow, and thermo-optical performance of nanofluids are caused by a variety of enhancing mechanisms and flow conditions. Despite the fact that a vast amount of study data has been documented in the literature, some outcomes are conflicting.

Many influencing aspects, as well as nonlinearity and refutations, make nanofluid research extremely difficult and obstruct its potentially useful uses. Machine learning methods, on the other hand, would be extremely valuable in nanofluid research for predicting thermophysical parameters, evaluating thermo-hydrodynamic performance, and evaluating radiative-optical performance in heat exchangers and solar energy systems [14].

In industrial and engineering applications where parts are cooked in ovens, a physics-informed neural network is designed to solve conductive heat transfer partial differential equations (PDEs), as well as convective heat transfer PDEs as boundary conditions (BCs). Current analysis methods based on trial-and-error finite element (FE) simulations are slow since convective coefficients are often unknown.

To meet the PDE, BCs, and initial condition, the loss function is defined based on mistakes. To reduce loss terms simultaneously, an adaptive normalizing system is designed. In addition, feature engineering employs heat transfer theory. By comparing 1D and 2D predictions to FE findings, the predictions for 1D and 2D situations are validated. When compared to theory-agnostic ML approaches, it is shown that only physics-informed activation functions can reliably predict heat transport beyond the training zone. Trained models were effectively utilized to evaluate the thermal responses of parts exposed to a wide range of temperatures in real time [15].

The influence of air injection and transverse baffles in shell and tube heat exchangers on thermohydraulic performance is predicted using enhanced prediction methods based on supervised machine-learning algorithms. The injection process involves injecting air into the shell at various flow rates in order to achieve the best thermohydraulic performance. To circumvent mathematical modeling or costly trials, four distinct machine-learning methods were used to forecast the heat exchanger's thermohydraulic performance. Random vector functional link, support vector machine, social media optimization, and k-nearest neighbors algorithm are among these algorithms. Experimental data was used to train and test the algorithms. The cold fluid and injected air volume flow rates were the algorithms' inputs, while the hot and cold fluid outlet temperatures, as well as pressure drop across the heat exchanger, were the algorithms' outputs [16].

In refrigeration systems, evaporative condensers are highly efficient and environmentally beneficial heat exchangers. Data-driven methods can help forecast the performance of evaporative condensers without the need for complicated theoretical study. Four machine learning models were used in this study to predict the heat transfer rate and overall heat transfer coefficient of a small-scale evaporative condenser operating under a wide range of working conditions: multi-layer perceptron artificial neural network (ANNMLP), support vector regression (SVR), decision tree (DT), and random forest (RF). The major affecting parameters in a set of experimental studies were intake air dry/wet-bulb temperatures, spraying water and condenser saturation temperatures, refrigerant, and air flow rates [17].

Heat convection and entropy generation in a hybrid nanofluid (Al2O3–Cu–water) flowing around a cylinder embedded in porous media were studied numerically.

For predictive analysis, an artificial neural network is employed, in which numerical data is generated to train an intelligence algorithm and optimize prediction mistakes. The heat transmission of the system increases as the Reynolds number, permeability parameter, or volume fraction of nanoparticles increases, according to the findings. The functional forms of these dependencies, on the other hand, are complicated.

The effect of increasing nanoparticle concentration on entropy generation is found to be nonmonotonic. To establish correlations for the shear stress and Nusselt number, particle swarm optimization is applied to the simulated and forecasted data. The ability of artificial intelligence algorithms to forecast the thermohydraulics and thermodynamics of thermal and solutal systems is demonstrated in this work [18].

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

II. EXPERIMENTAL PROCEDURE

Firstly, the data was written in the form Excel worksheet as shown in the below Figure 1.

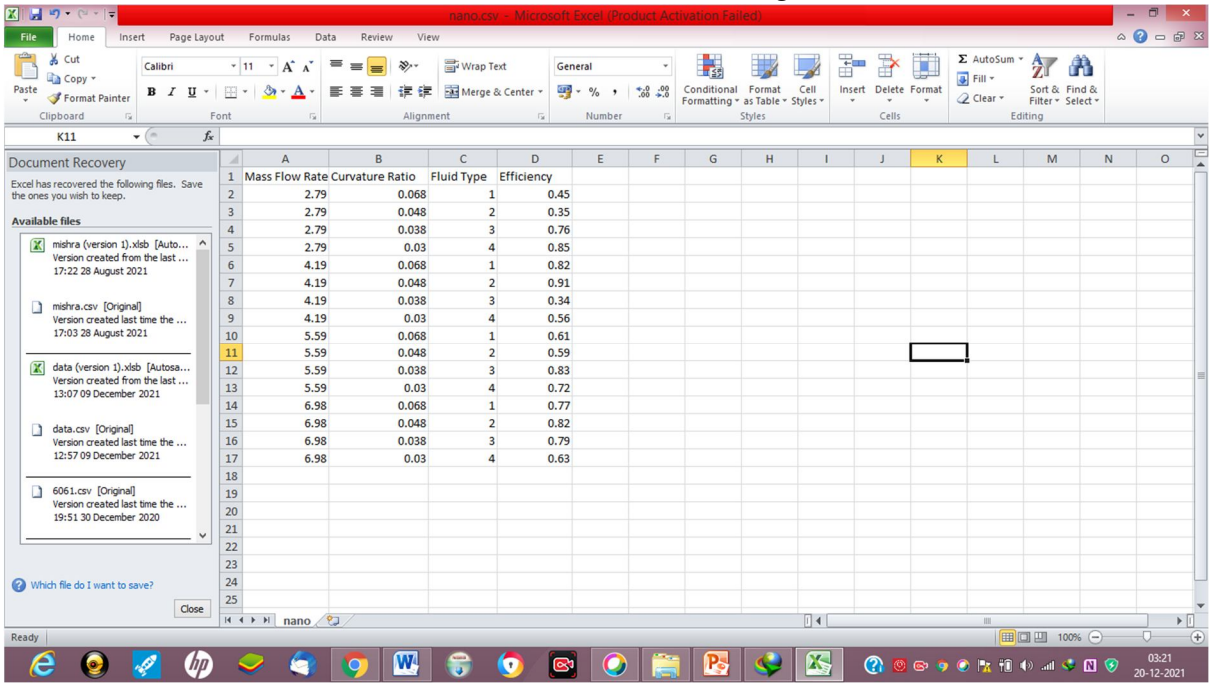


Figure 1: Data used in the present study

Data worksheet was saved in the form of csv file in the computer system. The saved data is later opened in the Google Colaboratory platform for subjecting it to given Machine Learning algorithms. Google colabratory platform is shown in Figure 2. Click on File option as shown in Figure 3 and click on “New Notebook”.

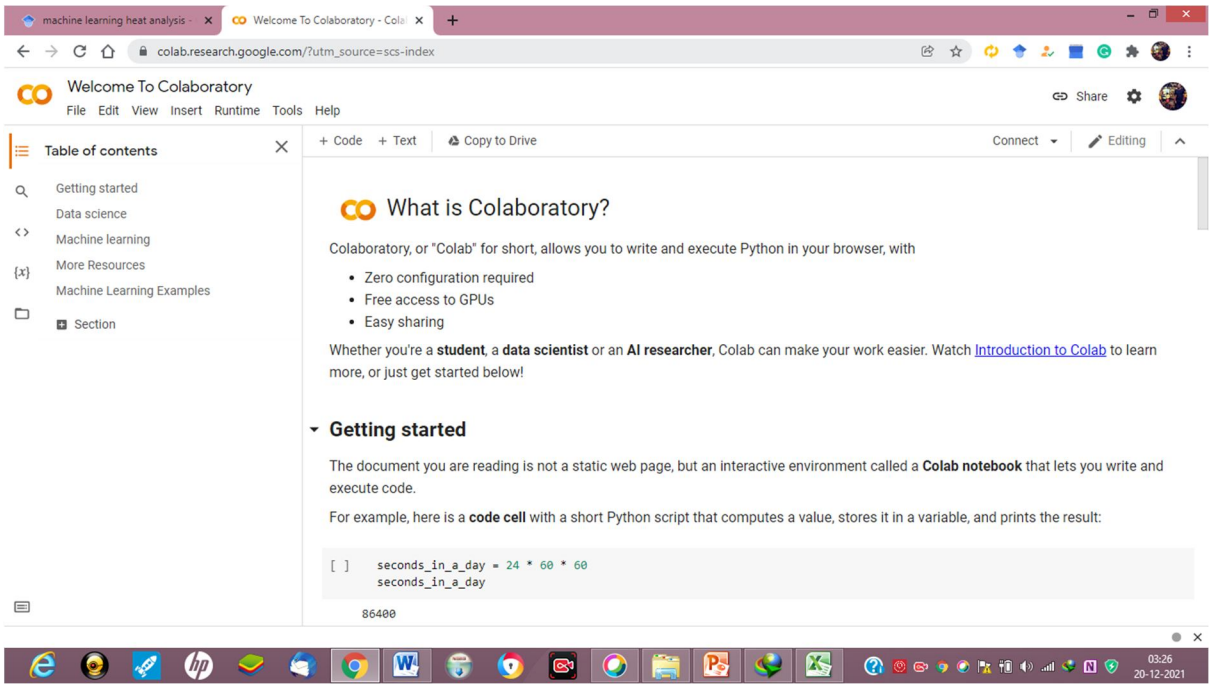


Figure 2: Google Colaboratory Platform

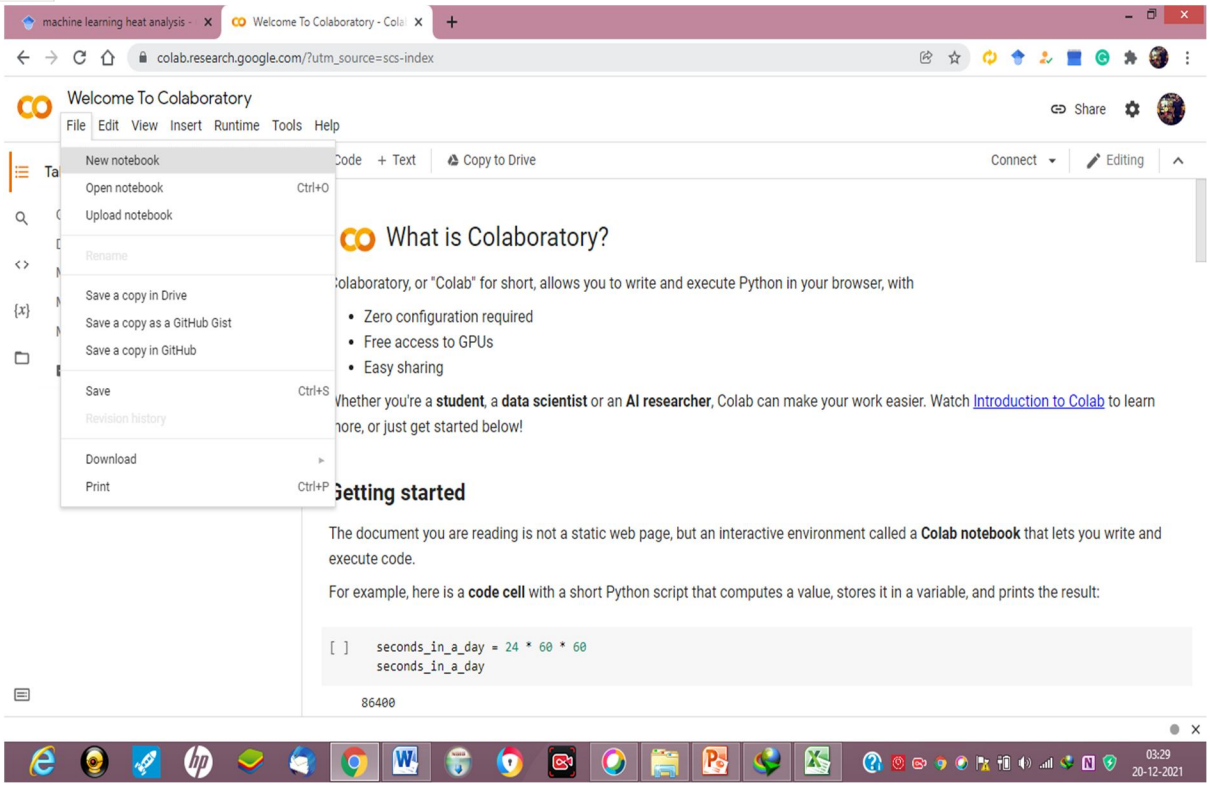


Figure 3: Opening New Notebook

Figure 4 shows the working environment where the Python coding is used to execute the machine learning algorithms.

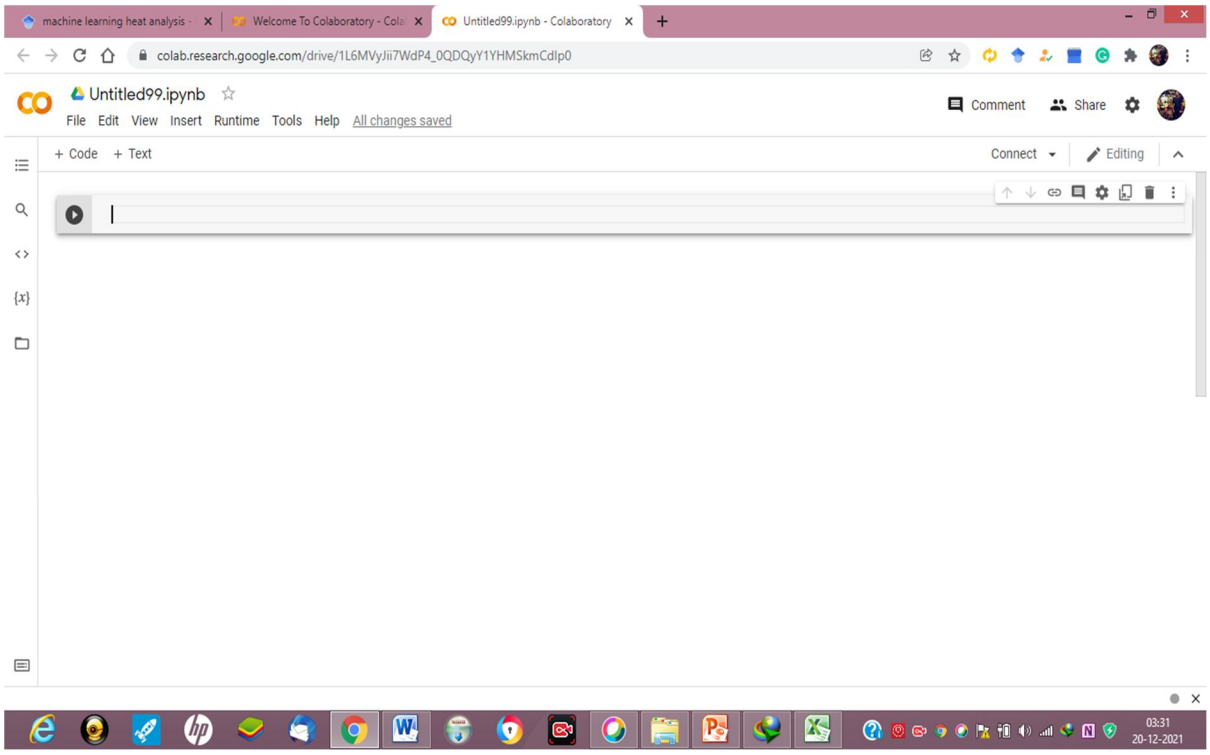


Figure 4: Coding Environment

In order to load the data csv file to the coding environment click on File icon as shown in Figure 5.

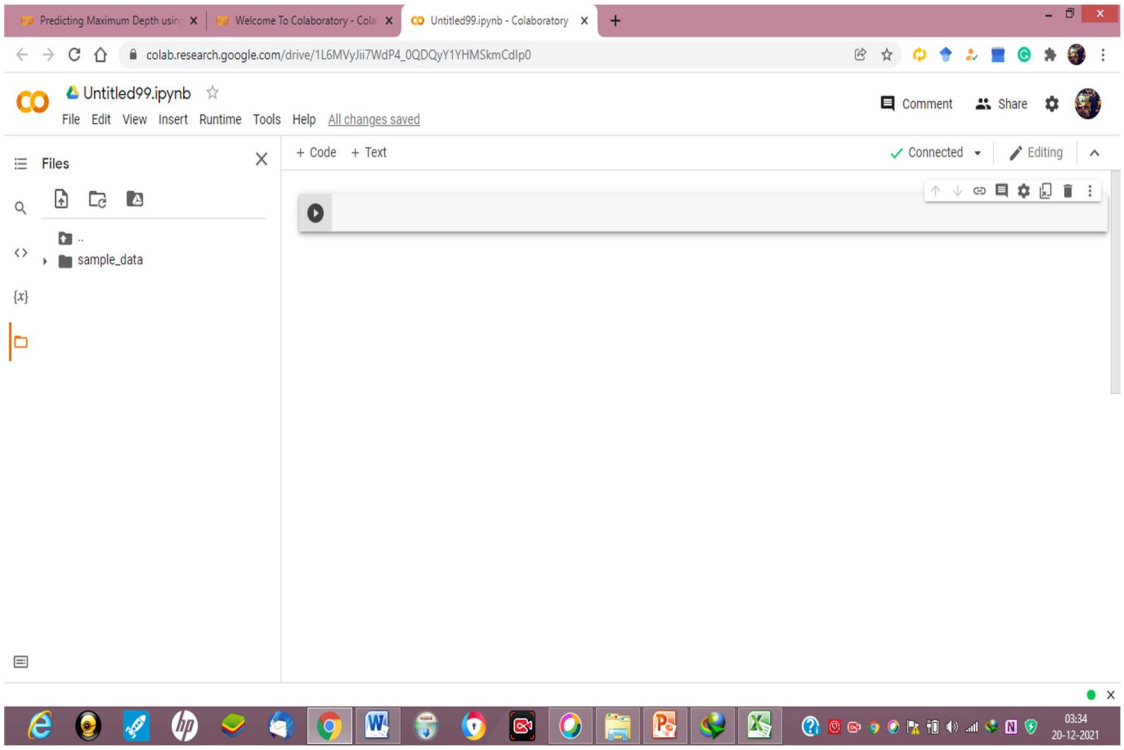


Figure 5: Click on File icon

Now click on upload icon to import the file as shown in Figure 6 and Figure 7 shows the imported file.

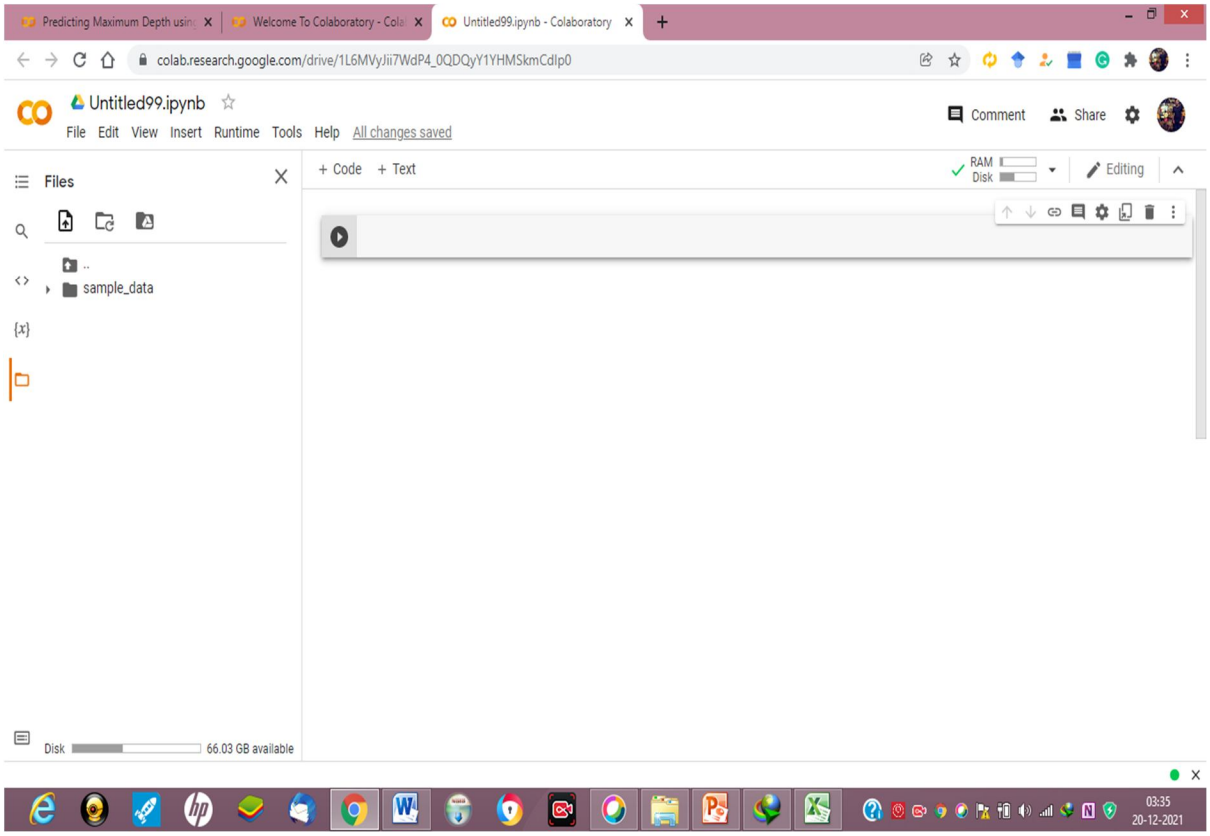


Figure 6: Click on upload icon and select the file to be uploaded

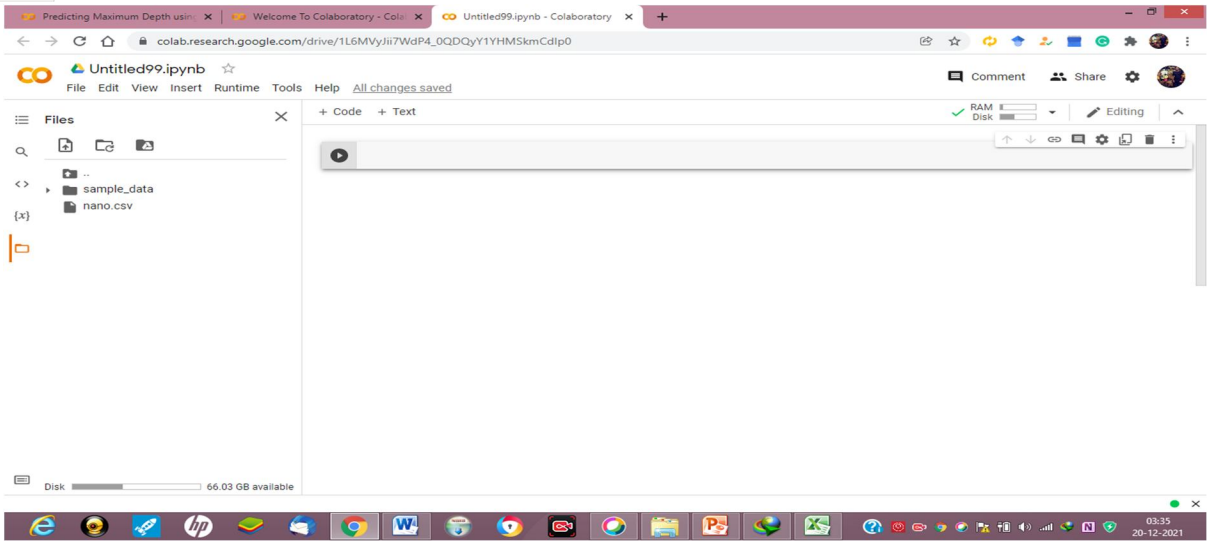


Figure 7: Imported data file

Now the below code is written to import the necessary python libraries to perform mathematical operations.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style("whitegrid")
plt.style.use("fivethirtyeight")
```

The features from the file is loaded by below python code.

```
nano= pd.read_csv('nano.csv')
```

In order to display the content of data below python code is used and the result obtained is shown in Figure 8.

```
nano.head( )
```

1 to 5 of 5 entries Filter

index	Mass Flow Rate	Curvature Ratio	Fluid Type	Efficiency
0	2.79	0.068	1	0.45
1	2.79	0.048	2	0.35
2	2.79	0.038	3	0.76
3	2.79	0.03	4	0.85
4	4.19	0.068	1	0.82

Show 10 per page
Like what you see? Visit the data table notebook to learn more about interactive tables.

Figure 8: Obtained results

nano.info() command shows the data types and number of missing values of the features used in present study as shown in Figure 9.

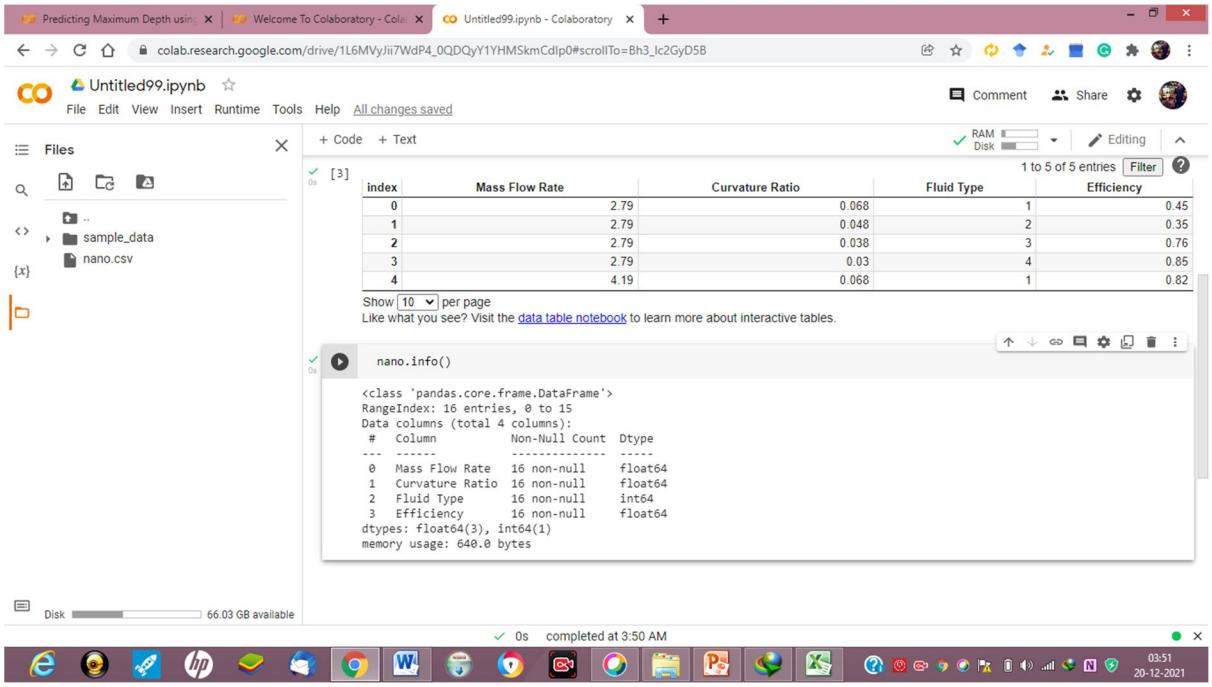


Figure 9: Data types and computation of null values

The statistical features is analysed by using *nano.describe()* command as shown in Figure 10.

1 to 8 of 8 entries Filter

index	Mass Flow Rate	Curvature Ratio	Fluid Type	Efficiency
count	16.0	16.0	16.0	16.0
mean	4.887500000000002	0.04600000000000001	2.5	0.6749999999999999
std	1.6131191317857876	0.014678782874157748	1.1547005383792515	0.1786243731036352
min	2.79	0.03	1.0	0.34
25%	3.84	0.036	1.75	0.5825
50%	4.890000000000001	0.043	2.5	0.74
75%	5.9375	0.05300000000000005	3.25	0.82
max	6.98	0.068	4.0	0.91

Show 25 per page
Like what you see? Visit the data table notebook to learn more about interactive tables.

Figure 10: Statistical features analysis

The input features (X) and the output feature (y) are defined by using below python coding.

```
X = nano[['Mass Flow Rate', 'Curvature Ratio', 'Fluid Type']]
y = nano[['Efficiency']]
```

The data set is divided into two sets i.e. training and testing set basically in 80-20 ratio. The code is defined below:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In order to measure the performance of the used Machine learning, metrics features are imported by using below code:

```
from sklearn import metrics
from sklearn.model_selection import cross_val_score

def cross_val(model):
    pred = cross_val_score(model, X, y, cv=10)
    return pred.mean()

def print_evaluate(true, predicted):
    mae = metrics.mean_absolute_error(true, predicted)
    mse = metrics.mean_squared_error(true, predicted)
    rmse = np.sqrt(metrics.mean_squared_error(true, predicted))
    r2_square = metrics.r2_score(true, predicted)
    print('MAE:', mae)
    print('MSE:', mse)
    print('RMSE:', rmse)
    print('R2 Square', r2_square)
    print('_____')

def evaluate(true, predicted):
    mae = metrics.mean_absolute_error(true, predicted)
    mse = metrics.mean_squared_error(true, predicted)
    rmse = np.sqrt(metrics.mean_squared_error(true, predicted))
    r2_square = metrics.r2_score(true, predicted)
    return mae, mse, rmse, r2_square
```

Standard scaling is done on the dataset by using below python code:

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

pipeline = Pipeline([
    ('std_scalar', StandardScaler())
])
X_train = pipeline.fit_transform(X_train)
X_test = pipeline.transform(X_test)
```

In present work, two machine learning algorithms are used i.e. Support Vector Machines algorithm and Random Forest algorithm.

Python code for implementing the Support Vector Machine is below:

```
from sklearn.svm import SVR

svm_reg = SVR(kernel='rbf', C=1000000, epsilon=0.001)
svm_reg.fit(X_train, y_train)

test_pred = svm_reg.predict(X_test)
train_pred = svm_reg.predict(X_train)
```

```
print('Test set evaluation:\n_____')
print_evaluate(y_test, test_pred)

print('Train set evaluation:\n_____')
print_evaluate(y_train, train_pred)
```

Python code for implementing the Random Forest algorithm is following:

```
from sklearn.ensemble import RandomForestRegressor

rf_reg = RandomForestRegressor(n_estimators=1000)
rf_reg.fit(X_train, y_train)

test_pred = rf_reg.predict(X_test)
train_pred = rf_reg.predict(X_train)

print('Test set evaluation:\n_____')
print_evaluate(y_test, test_pred)

print('Train set evaluation:\n_____')
print_evaluate(y_train, train_pred)
```

III. RESULTS AND DISCUSSION

A. Support Vector Machines

The "Support Vector Machine" (SVM) is a supervised machine learning technique that can solve classification and regression problems. It is, however, mostly employed to solve categorization difficulties. Each data item is plotted as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a certain coordinate in the SVM algorithm as shown in Figure 11. Then we accomplish classification by locating the hyper-plane that clearly distinguishes the two classes. Simply put, support vectors are the coordinates of each individual observation. The SVM classifier is a frontier that separates the two classes (hyper-plane/line) the most effectively.

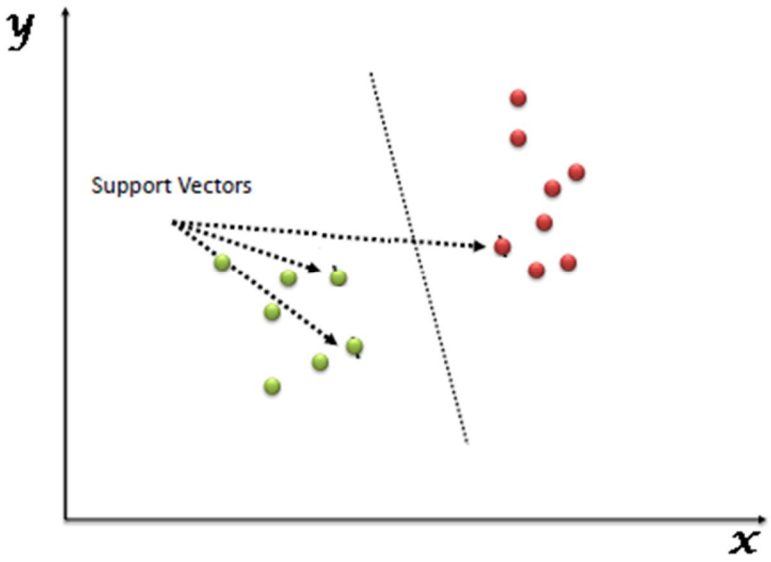


Figure 11: Representation of Support Vector Machine algorithm

It is simple to create a linear hyper-plane between these two classes in the SVM classifier. However, another pressing concern is whether we should need to manually implement this feature in order to have a hyper-plane. The kernel trick is a technique used by the SVM algorithm. The SVM kernel is a function that changes a not separable problem into a separable problem by taking a low-dimensional input space and transforming it to a higher-dimensional space. It is most beneficial in problems with non-linear separation. Simply simply, it does some fairly sophisticated data transformations before determining how to segregate the data using the labels or outputs you've specified.

The results obtained are shown in Figure 12.

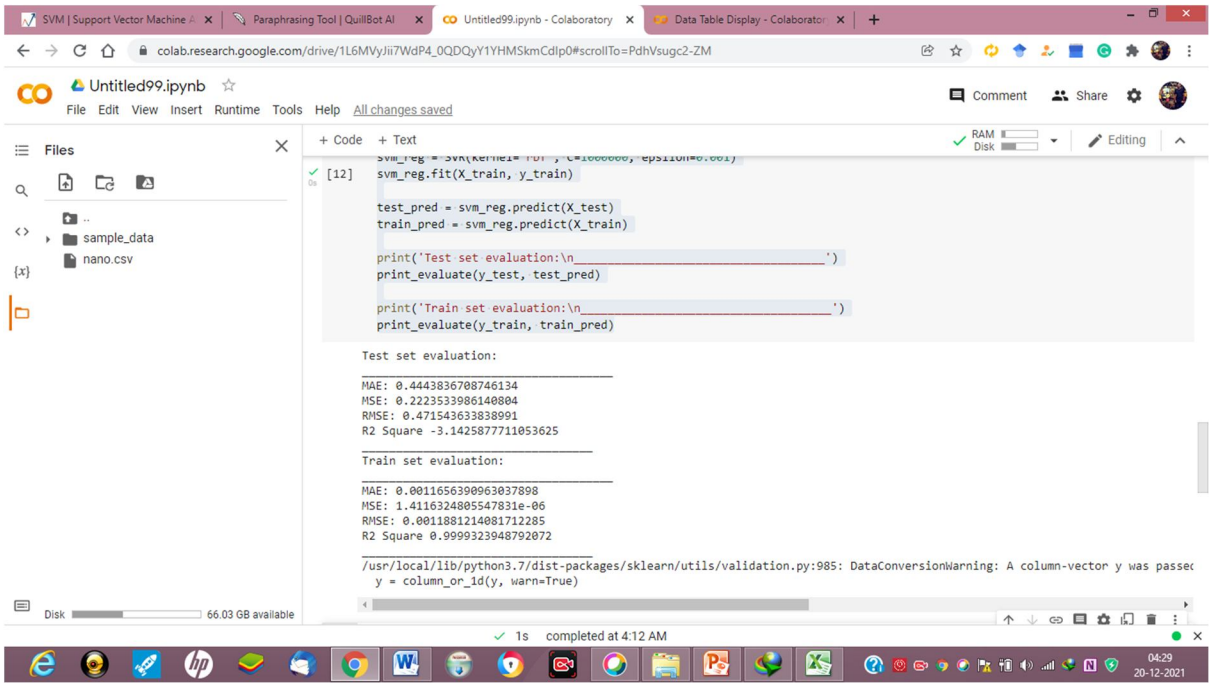


Figure 12: Results obtained by Support Vector Machine algorithm

In a simple way the results obtained by Support Vector Machine is following:

Test set evaluation:

_____

MAE: 0.4443836708746134
MSE: 0.2223533986140804
RMSE: 0.471543633838991
R2 Square -3.1425877711053625

_____

Train set evaluation:

_____

MAE: 0.0011656390963037898
MSE: 1.4116324805547831e-06
RMSE: 0.0011881214081712285
R2 Square 0.9999323948792072

_____

/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:985: DataConversionWarning: 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().
y = column_or_1d(y, warn=True)

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

B. Random Forest Algorithm

Random forest is a supervised machine learning algorithm that is commonly used to solve classification and regression problems. It creates decision trees from various samples, using the majority vote for classification and the average for regression.

One of the most essential characteristics of the Random Forest Algorithm is that it can handle data sets with both continuous and categorical variables, as in regression and classification. For classification difficulties, it produces superior results.

A random forest is a machine learning technique for solving classification and regression problems. It makes use of ensemble learning, which is a technique for solving complicated problems by combining several classifiers.

Many decision trees make up a random forest algorithm. Bagging or bootstrap aggregation are used to train the 'forest' formed by the random forest method. Bagging is a meta-algorithm that increases the accuracy of machine learning methods by grouping them together.

The (random forest) algorithm determines the outcome based on decision tree predictions. It forecasts by averaging or averaging the output of various trees. The precision of the result improves as the number of trees grows.

A random forest method overcomes the drawbacks of a decision tree algorithm. It reduces dataset overfitting and improves precision. It generates forecasts without requiring a large number of package setups (like scikit-learn).

A random forest algorithm's building components are decision trees. A decision tree is a decision-making tool with a tree-like structure. A basic understanding of decision trees will aid our understanding of random forest algorithms.

There are three parts to a decision tree: decision nodes, leaf nodes, and a root node. A decision tree method separates a training dataset into branches, each of which is further divided into branches. This pattern repeats until a leaf node is reached. There is no way to separate the leaf node any farther.

The attributes utilized to forecast the outcome are represented by the nodes in the decision tree. The leaves are connected to the decision nodes. The three types of nodes in a decision tree are depicted in the Figure 13.

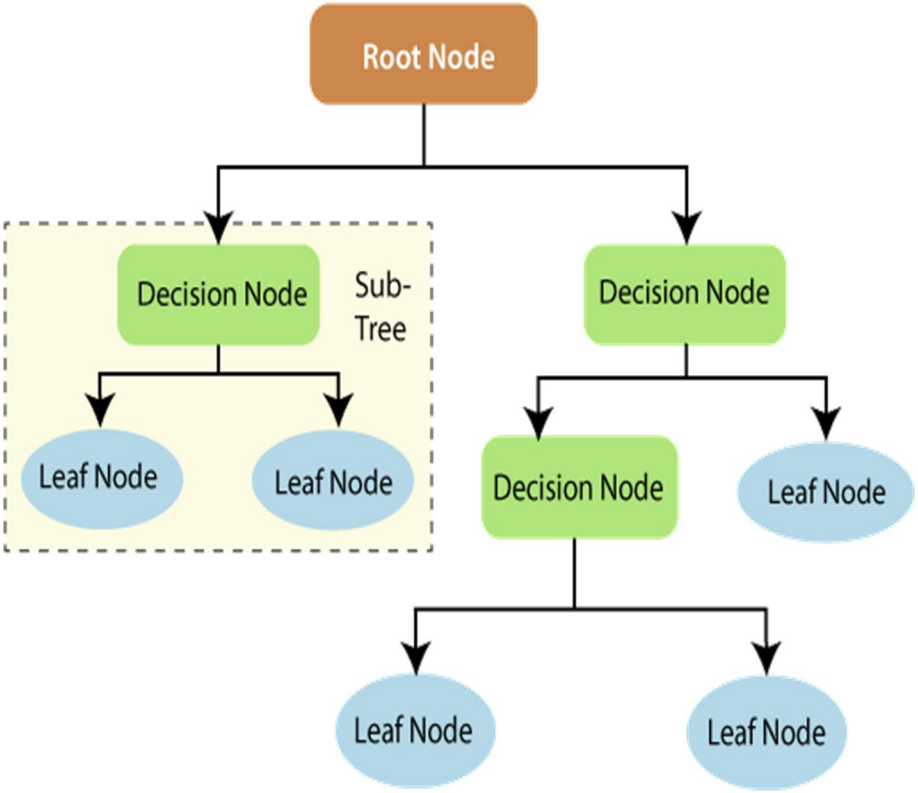


Figure 13: Nodes in Decision trees

More information on how decision trees work can be found in information theory. The basic blocks of decision trees are entropy and information gain. An knowledge of how decision trees are constructed will benefit from a review of these key ideas.

The measure entropy is used to calculate uncertainty. Given a set of independent variables, information gain is a measure of how much uncertainty in the target variable is decreased.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

The notion of information gain entails employing independent variables (features) to learn about a target variable (class). The information gain is calculated using the entropy of the target variable (Y) and the conditional entropy of Y (given X). The conditional entropy is deducted from the entropy of Y in this scenario.

In the training of decision trees, information gain is used. It contributes to the reduction of uncertainty in these trees. A high degree of uncertainty (information entropy) has been reduced with a substantial information gain. Splitting branches, a fundamental action in the creation of decision trees, requires entropy and information gain.

Let's look at how a decision tree works in practice. Let's say we want to anticipate whether a customer will buy a phone or not. His judgment is based on the phone's specifications.

The features of the phone are represented by the root node and decision nodes of the decision. The final output, whether buying or not buying, is represented by the leaf node. The pricing, internal storage, and Random Access Memory are the primary factors that influence the decision (RAM).

The fundamental distinction between the decision tree and the random forest algorithms is that the latter randomly establishes root nodes and segregates nodes. The bagging method is used by the random forest to generate the required forecast.

Rather than using a single sample of data, bagging includes using many samples (training data). A training dataset is a collection of observations and attributes used to make predictions. Depending on the training data provided to the random forest algorithm, the decision trees produce varied results. These outputs will be ranked, and the one with the best score will be chosen as the final product.

Random forest classification uses an ensemble methodology to achieve the desired result. Various decision trees are trained using the training data. This dataset contains observations and features that will be chosen at random when nodes are split.

Various decision trees are used in a rain forest system. There are three types of nodes in a decision tree: decision nodes, leaf nodes, and the root node. Each tree's leaf node represents the final output produced by that particular decision tree. The final product is chosen using a majority-voting procedure. In this situation, the final output of the rain forest system is the output chosen by the majority of decision trees. A simple random forest classifier is depicted in the Figure 14.

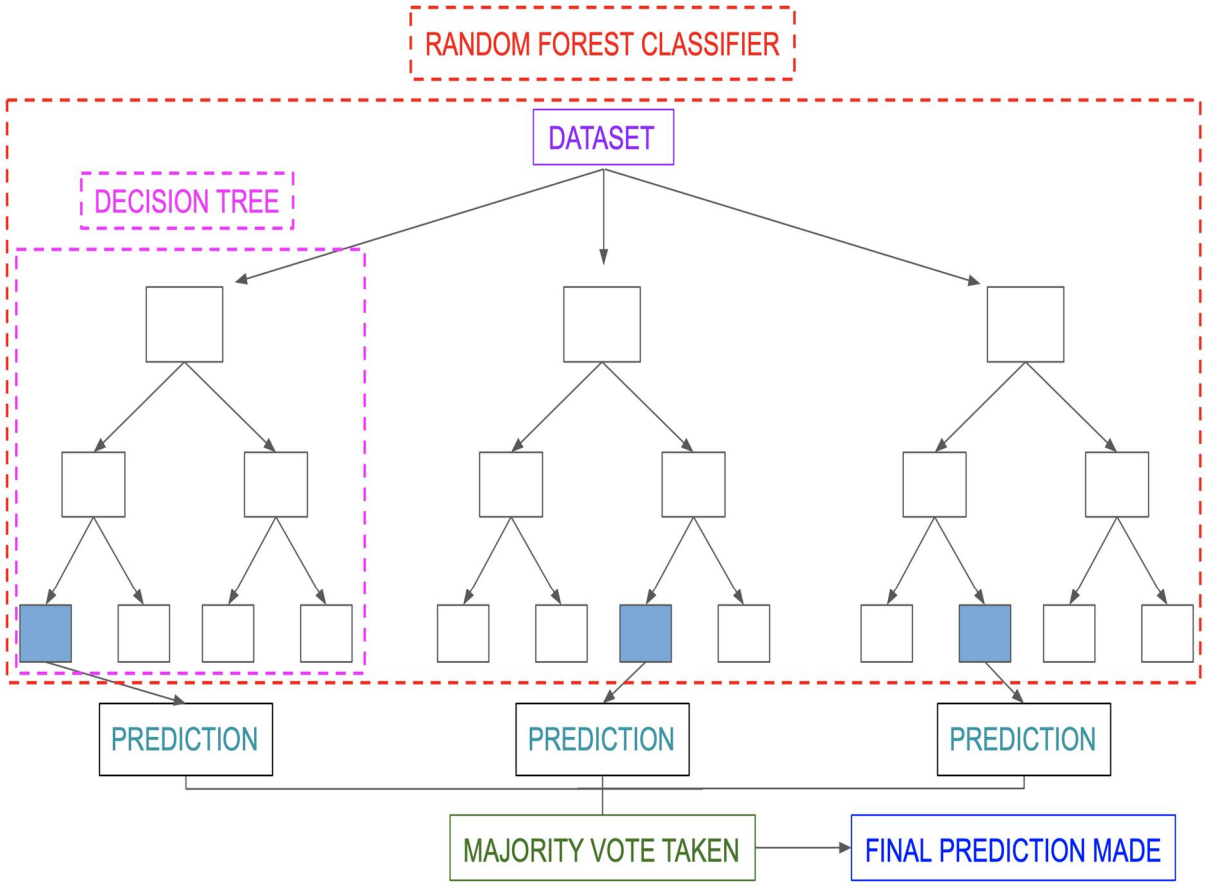


Figure 14: Representation of the random forest classifier

The output obtained in the current study is shown in Figure 15.

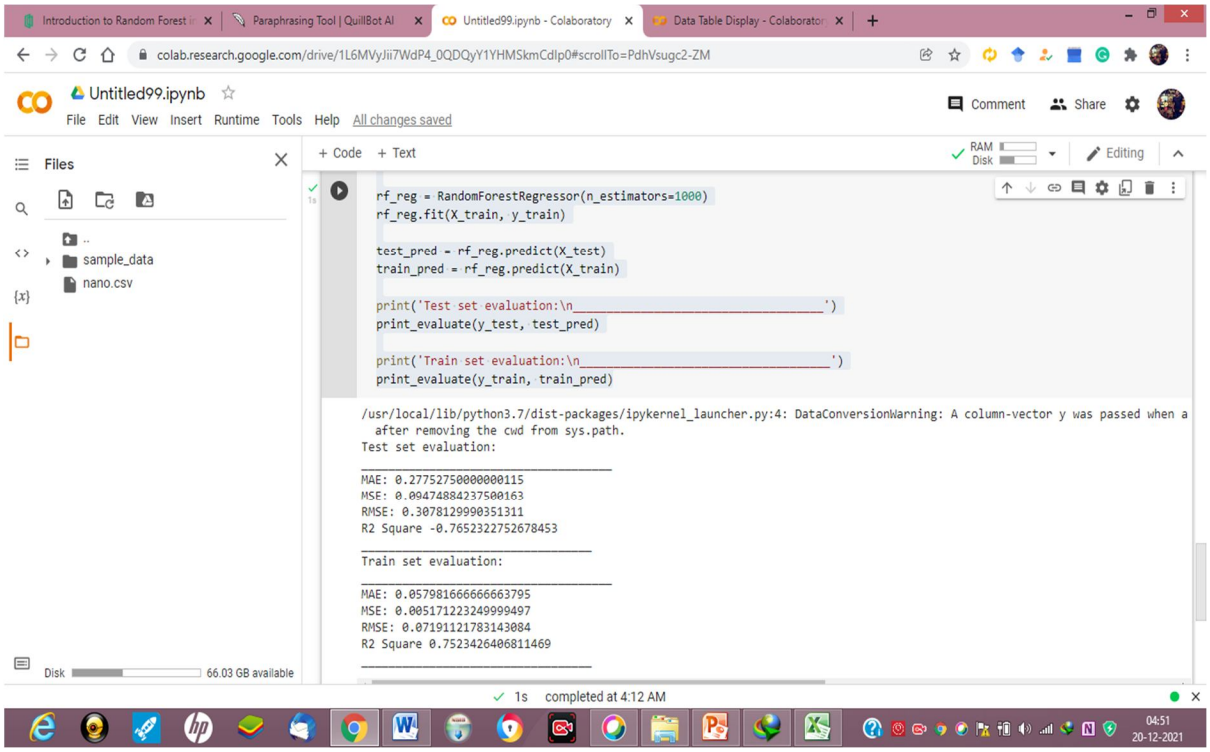


Figure 15: Output obtained from random forest algorithm

The results obtained are below:

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: DataConversionWarning: 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().

after removing the cwd from sys.path.

Test set evaluation:

_____

MAE: 0.27752750000000115
MSE: 0.09474884237500163
RMSE: 0.3078129990351311
R2 Square -0.7652322752678453

_____

Train set evaluation:

_____

MAE: 0.057981666666663795
MSE: 0.005171223249999497
RMSE: 0.07191121783143084
R2 Square 0.7523426406811469

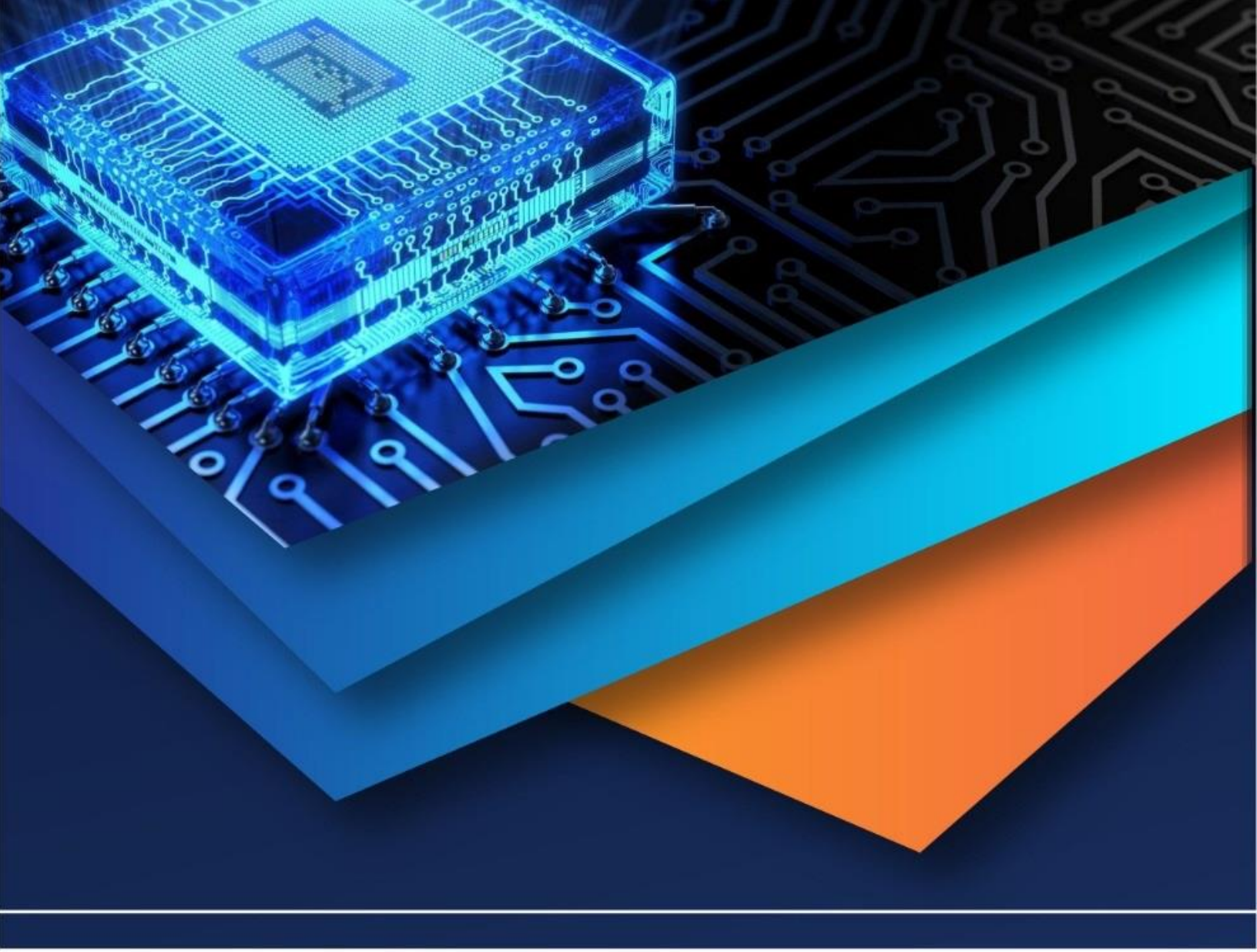
_____

IV. CONCLUSIONS

Machine Learning algorithms were successfully implemented in the given dataset. After comparing the metrics features value of MAE, MSE and R-square value it is concluded that Random Forest algorithm has more accuracy in comparison to the Support Vector Machine algorithm. In order to improve the performance of the machine learning algorithms more dataset should be taken into account and the future study of this work is to implement other types of supervised and unsupervised machine learning algorithms.

REFERENCES

[1] Goodfellow, I., Bengio, Y. and Courville, A., 2016. Machine learning basics. Deep learning, 1(7), pp.98-164.
[2] Mohri, M., Rostamizadeh, A. and Talwalkar, A., 2018. Foundations of machine learning. MIT press.
[3] Jordan, M.I. and Mitchell, T.M., 2015. Machine learning: Trends, perspectives, and prospects. Science, 349(6245), pp.255-260.
[4] Sammut, C. and Webb, G.I. eds., 2011. Encyclopedia of machine learning. Springer Science & Business Media.
[5] Thapliyal, S. and Mishra, A., 2021. Machine learning classification-based approach for mechanical properties of friction stir welding of copper. Manufacturing Letters, 29, pp.52-55.
[6] Mishra, A. and Dixit, D., 2021. Brain Inspired Computing Approach for the Optimization of the Thin Film Thickness of Polystyrene on the Glass Substrates. arXiv preprint arXiv:2107.12156.
[7] Mishra, A., 2020. Local binary pattern for the evaluation of surface quality of dissimilar Friction Stir Welded Ultrafine Grained 1050 and 6061-T6 Aluminium Alloys.
[8] Mishra, A. and Patti, A., 2021. Deep Convolutional Neural Network Modeling and Laplace Transformation Algorithm for the Analysis of Surface Quality of Friction Stir Welded Joints.
[9] Mishra, A., Tiwari, A. and Dubey, N.K., 2020. Machine Learning Approach to Determine Corrosion Potential of Friction Stir Welded Joints. Journal of Image Processing & Pattern Recognition Progress, 7(1), pp.5-17.
[10] Mishra, A., 2020. Artificial Intelligence Algorithms for the Analysis of Mechanical Property of Friction Stir Welded Joints by using Python Programming. Welding Technology Review, 92.
[11] Shi, Zenan, Xueying Yuan, Yaling Yan, Yuanlin Tang, Junjie Li, Hong Liang, Lianpeng Tong, and Zhiwei Qiao. "Techno-economic analysis of metal–organic frameworks for adsorption heat pumps/chillers: from directional computational screening, machine learning to experiment." Journal of Materials Chemistry A 9, no. 12 (2021): 7656-7666.
[12] Parida, D.R., Dani, N. and Basu, S., 2021. Data-driven analysis of molten-salt nanofluids for specific heat enhancement using unsupervised machine learning methodologies. Solar Energy, 227, pp.447-456.
[13] Mohammadpour, J., Husain, S., Salehi, F. and Lee, A., 2022. Machine learning regression-CFD models for the nanofluid heat transfer of a microchannel heat sink with double synthetic jets. International Communications in Heat and Mass Transfer, 130, p.105808.
[14] Ma, T., Guo, Z., Lin, M. and Wang, Q., 2021. Recent trends on nanofluid heat transfer machine learning research applied to renewable energy. Renewable and Sustainable Energy Reviews, 138, p.110494.
[15] Zobeiry, N. and Humfeld, K.D., 2021. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. Engineering Applications of Artificial Intelligence, 101, p.104232.
[16] El-Said, E.M., Abd Elaziz, M. and Elsheikh, A.H., 2021. Machine learning algorithms for improving the prediction of air injection effect on the thermohydraulic performance of shell and tube heat exchanger. Applied Thermal Engineering, 185, p.116471.
[17] Behnam, P., Faegh, M., Shafii, M.B. and Khiadani, M., 2021. A comparative study of various machine learning methods for performance prediction of an evaporative condenser. International Journal of Refrigeration, 126, pp.280-290.
[18] Tang, X., Guo, Q., Li, M., Wei, C., Pan, Z. and Wang, Y., 2021. Performance analysis on liquid-cooled battery thermal management for electric vehicles based on machine learning. Journal of Power Sources, 494, p.229727.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY