3D PRINTER MATERIAL PREDICTION USING IBM WATSON

MINI PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

KOUDAGANI GANESH	**19UK1A05H0**
ARUKALA KAVYA	**19UK1A05G7**
SUVARNA GNANA PRASANNA	**19UK1A05L5**
RAGAM MAHENDER	**19UK1A05J4**

Under the esteemed guidance of

Mr.P. ILANNA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal – 506005

2019– 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA, WARANGAL – 506005

2019– 2023

<u>CERTIFICATE OF COMPLETION</u>

<u>UG PROJECT PHASE-1</u>

This is to certify that the UG Project Phase-1 entitled**"3D PRINTER MATERIAL PREDICTION USING IBM WATSON"**is being submitted by **KOUDAGANI GANESH(19UK1A05H0),SUVARNA GNANA PRASANNA(19UK1A05L5),ARUKALA KAVYA(19UK1A05G7),RAGAM MAHENDER(19UK1A05J4)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering to Jawaharlal Nehru Technological University Hyderabad** during the academic year **2022-23,** is a record of work carried out by them under the guidance and supervision.

Project Guide	**Head of the Department**
Mr.P.ILANNA	**Dr. R. Naveen Kumar**
(Assistant Professor)	(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.We extend our heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.Phase-1. We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG ProjectWe express heartfelt thanks to the guide, **Mr. P. ILANNA** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

KOUDAGANI GANESH	**19UK1A05H0**
ARUKALA KAVYA	**19UK1A05G7**
SUVARNA GNANA PRASANNA	**19UK1A05L5**
RAGAM MAHENDER	**19UK1A05J4**

TABLE OF CONTENTS: -

3D PRINTING MATERIAL DECTECTION USING IBM WATSON

1.INTRODUCTION

1.1OVERVIEW

The 3D printing materials industry is increasing due to the rise in the demand from healthcare, automotive, and other industries, globally. The 3D printing materials market comprises several stakeholders, such as raw material suppliers, processors, end-product manufacturers, and regulatory organizations in the supply chain. The demand side of this market is characterized by the development of various industries such as aerospace & defense, healthcare, consumer goods, and automotive. Advancements in technology and diverse applications characterize the supply side. Various primary sources from both the supply and demand sides of the market were interviewed to obtain qualitative and quantitative information.

1.2PURPOSE

Predicting material would be more suitable for making the 3D model. In this project, the input parameters are like Layer Height (mm), Wall Thickness (mm), Infill Density (%), Infill Pattern (honeycomb, grid), Nozzle Temperature (Cº), Bed Temperature (Cº), Print Speed(mm/s), Fan Speed (%), Roughness (µm), Tension (ultimate), Strength (MPa), Elongation (%).
Based on these parameters a supervised machine learning model is built to predict the best material to be used for building 3D models. A web application is build so that the user can type in the mentioned parameters and the material which suits the best is showcased on UI

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

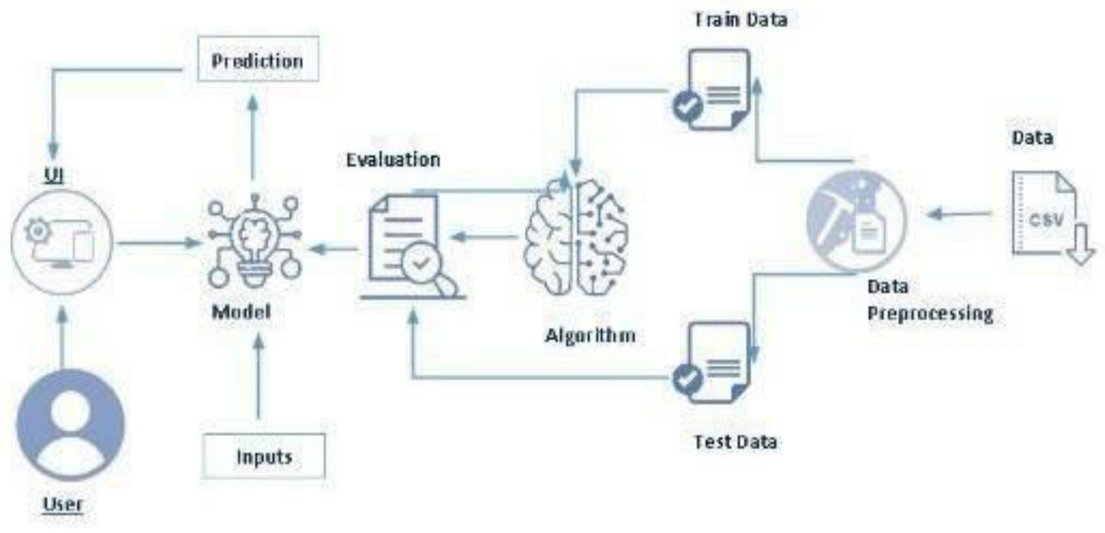
While 3D printing allows engineers to produce single items inexpensively, it sometimes comes at a cost to quality. Aside from high-end machines that costs millions of dollars to purchase, many 3D printers produce good that are inferior to those made through traditional manufacturing. One of the reasons for this is a lack of universal standards.
“Put simply, many manufacturers and end users have difficulty stating with certainty that parts or products produced via 3D printing—whether all on the same printer or across geographies—will be of consistent quality, strength, and reliability,” . “Without this guarantee, many manufacturers will remain leery of AM technology, judging the risks of uncertain quality to be too costly a trade-off for any gains they might realize.”

2.2 PROPOSED SYSTEM

The merging of artificial intelligence and 3D printing is an evolution of manufacturing paradigms. Prosthetics design, for instance, is one of the most important applications of 3D printing. As technology advances, artificial intelligence and 3D printers can be used to control 3D printers and increase the number of compatible materials for the process. By combining these two technologies, manufacturers can create new and improved products and production processes. Artificial intelligence and 3D printing will eventually help humans create better prosthetics.

3.THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE AND SOFTWARE DESIGNING

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum , and first released on February 20, 1991. Its high-level built in data structures, combined with dynamic typing and dynamic binding , make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on

Windows, Linux, and macOS. Conda is an open-source, crossplatform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use

Spyder

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.Spyder is extensible with first-party and third party plugins includes support for interactive tools for data inspection and embeds Pythonspecific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

Flask

Web frame work used for building. It is a web application framework written in python which will be running in local browser with a user interface. In this application, whenever the user interacts with UI and selects emoji, it will suggest the best and top movies of that genre to the use.

Hardware Requirements:

o Operating system: window 7 and above with 64bit

o Processor Type -Intel Core i3-3220

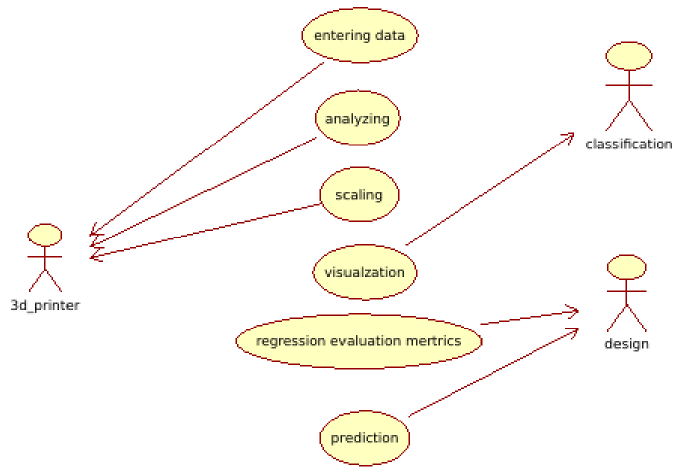
o RAM: 4Gb and above

o Hard disk: min 100GB

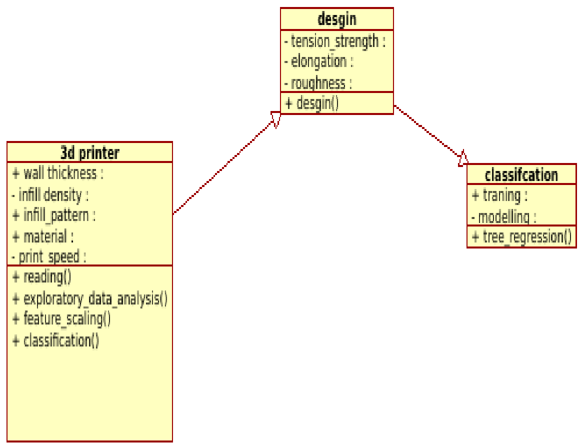
4.EXPERIMENTAL INVESTIGATION

Here we are going to build a machine learning model that predicts whether the given message is a spam or not, based on these parameters a supervised machine learning model is built to predict the best material to be used for building 3D models. A web application is build so that the user can type in the mentioned part a meters and the material which suits the best is showcased on UI.

5.FLOWCHART



USE CASE DIAGRAM



6.ADVANTAGES

- Easy to use
- Cost efficient
- Time efficient

7.CONCLUSION

3D printing technology could revolutionize and re-shape the world. Advance in 3D technology can significantly change and improve the way we manufacture products goods worldwide.

If the last industrial revolution brought us mass production and the advent of economics of scale – the digital 3D printing revolution could bring mass manufacturing back a full of circle – to an era of mass personalization, and return to individual craftsmanship.

8.FUTURE SCOPE

Future applications for 3D printing might include creating open-source scientific equipment to create opensource labs

Science-based applications like reconstructing fossils in paleontology . Replicating ancient and priceless artifacts in archaeology

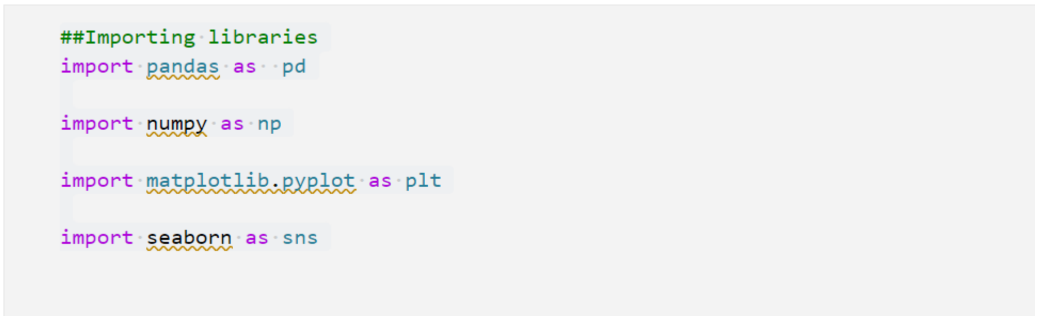
Reconstructing bones and body parts in forensic pathology.The technology currently being researched for building construction.

9.BIBILOGRAPHY

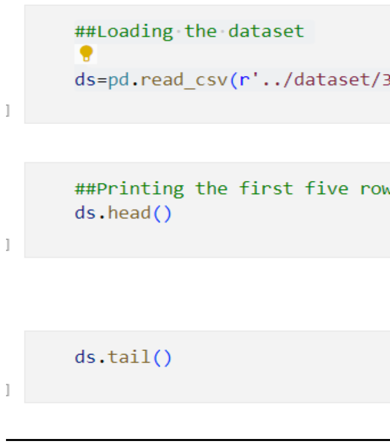
- http://mashable.com/2014/03/06/3d-printed-blood-vessels/
- http://www.3dprinter.net/referemce/what-is-3d-printing

10.APPENDIX

SOURCE CODE:



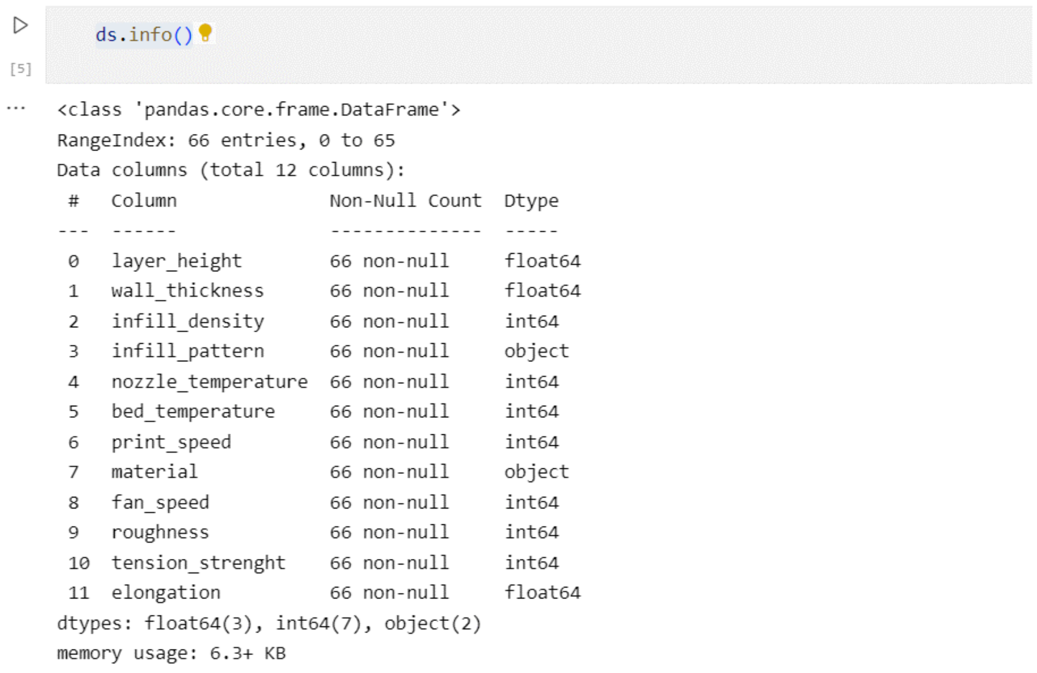
```
##Importing libraries
import pandas as  pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```



```
##Loading the dataset
ds=pd.read_csv(r'../dataset/3
```

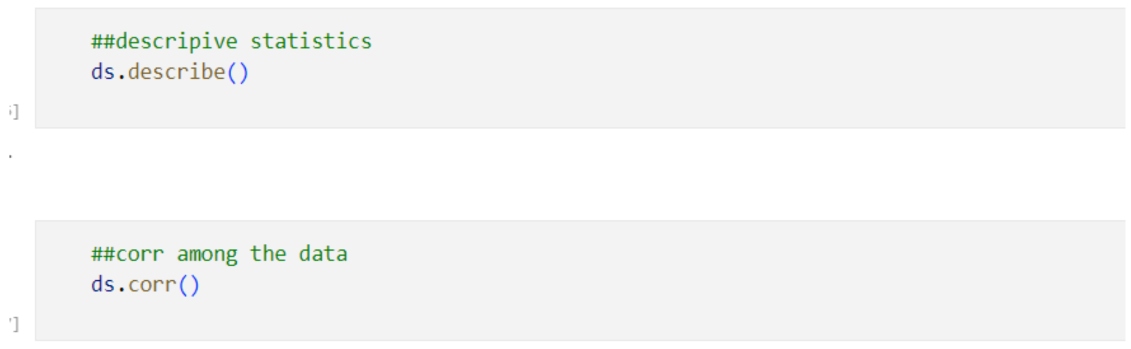
```
##Printing the first five row
ds.head()
```

```
ds.tail()
```



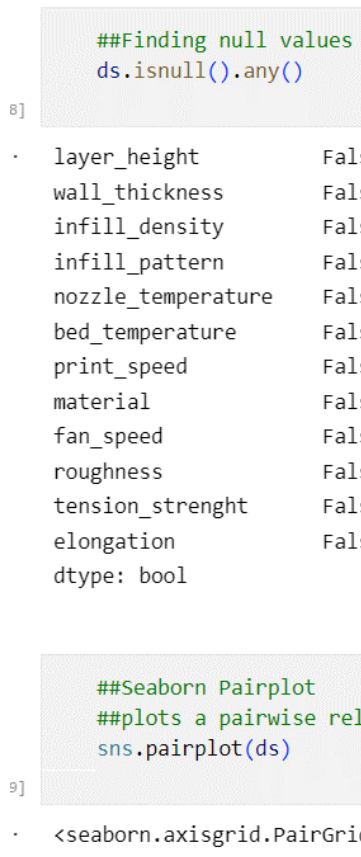
```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66 entries, 0 to 65
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   layer_height        66 non-null     float64
 1   wall_thickness      66 non-null     float64
 2   infill_density      66 non-null     int64
 3   infill_pattern      66 non-null     object
 4   nozzle_temperature  66 non-null     int64
 5   bed_temperature     66 non-null     int64
 6   print_speed         66 non-null     int64
 7   material            66 non-null     object
 8   fan_speed           66 non-null     int64
 9   roughness           66 non-null     int64
 10  tension_strenght    66 non-null     int64
 11  elongation          66 non-null     float64
dtypes: float64(3), int64(7), object(2)
memory usage: 6.3+ KB
```



```
##descripive statistics
ds.describe()
```

```
##corr among the data
ds.corr()
```

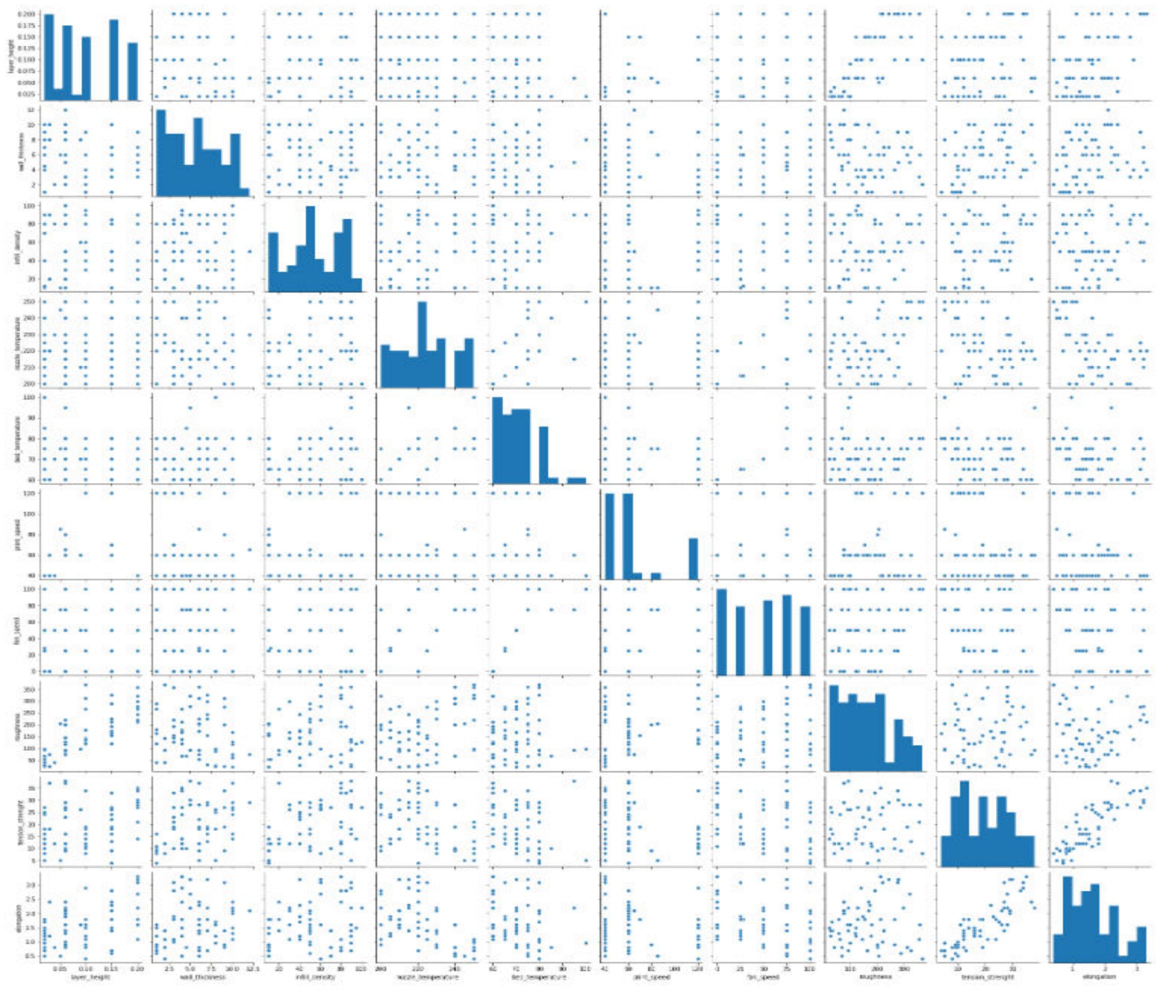


```
##Finding null values
ds.isnull().any()
```

```
layer_height          Fal
wall_thickness        Fal
infill_density        Fal
infill_pattern        Fal
nozzle_temperature    Fal
bed_temperature       Fal
print_speed           Fal
material              Fal
fan_speed             Fal
roughness             Fal
tension_strenght      Fal
elongation            Fal
dtype: bool
```

```
##Seaborn Pairplot
##plots a pairwise rel
sns.pairplot(ds)
```

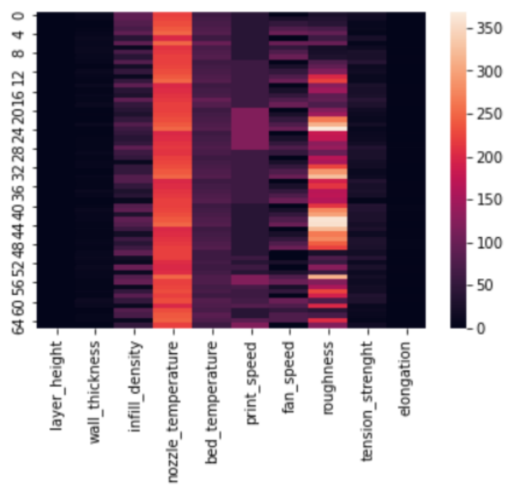
```
<seaborn.axisgrid.PairGri
```



```
##Seaborn Heatmap
##A way of representing the data in 2-D form
sns.heatmap(ds[['layer_height','wall_thickness','infill_density','nozzle_temperature','bed_temperature','print_speed','fan_speed','roughness','tension_strenght','elongation']])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27b306c9640>
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27b306c9640>
```



```python
##Label Emcoding
from sklearn.preprocessing import LabelEncoder

lb=LabelEncoder()

ds=ds.iloc[:,:].values
```

```python
ds[:,3]=lb.fit_transform(ds[:,3])

ds[:,7]=lb.fit_transform(ds[:,7])
```

```python
da=pd.DataFrame(ds)
```

```python
y=ds[:,7]

y=y.astype("int")
```

```python
da.drop(columns=7,inplace=True)
```

```python
x=da.iloc[:,:].values

x
```

```
Output exceeds the size limit. Open the full output data in a text editor
array([[0.02, 8.0, 90, 0, 220, 60, 40, 0, 25, 18, 1.2],
       [0.02, 7.0, 90, 1, 225, 65, 40, 25, 32, 16, 1.4],
       [0.02, 1.0, 80, 0, 230, 70, 40, 50, 40, 8, 0.8],
       [0.02, 4.0, 70, 1, 240, 75, 40, 75, 68, 10, 0.5],
       [0.02, 6.0, 90, 0, 250, 80, 40, 100, 92, 5, 0.7],
       [0.02, 10.0, 40, 1, 200, 60, 40, 0, 60, 24, 1.1],
       [0.02, 8.0, 90, 0, 250, 100, 40, 100, 98, 5, 0.95],
       [0.02, 10.0, 10, 1, 210, 70, 40, 50, 21, 14, 1.5],
       [0.02, 9.0, 70, 0, 215, 75, 40, 75, 24, 27, 1.4],
       [0.02, 8.0, 40, 1, 220, 80, 40, 100, 30, 25, 1.7],
       [0.06, 6.0, 80, 0, 220, 60, 60, 0, 75, 37, 2.4],
       [0.06, 2.0, 20, 1, 225, 65, 60, 25, 92, 12, 1.4],
       [0.06, 10.0, 50, 0, 230, 70, 60, 50, 118, 16, 1.3],
       [0.06, 6.0, 10, 1, 240, 75, 60, 75, 200, 9, 0.8],
       [0.06, 3.0, 50, 0, 250, 80, 60, 100, 220, 10, 1.0],
       [0.06, 10.0, 90, 1, 200, 60, 60, 0, 126, 27, 2.2],
       [0.06, 3.0, 40, 0, 205, 65, 60, 25, 145, 23, 1.9],
       [0.06, 8.0, 30, 1, 210, 70, 60, 50, 88, 26, 1.6],
       [0.06, 5.0, 90, 0, 215, 95, 60, 75, 92, 38, 2.2],
       [0.06, 10.0, 50, 1, 220, 80, 60, 100, 74, 29, 2.0],
       [0.1, 1.0, 40, 0, 220, 60, 120, 0, 120, 16, 1.2],
       [0.1, 2.0, 30, 1, 225, 65, 120, 25, 144, 12, 1.1],
       [0.1, 1.0, 50, 0, 230, 70, 120, 50, 265, 10, 0.9],
       [0.1, 9.0, 80, 1, 240, 75, 120, 75, 312, 19, 0.8],
       [0.1, 2.0, 60, 0, 250, 80, 120, 100, 368, 8, 0.4],
...
       [0.06, 9.0, 10, 1, 200, 75, 80, 75, 200, 9, 0.9],
       [0.04, 2.0, 80, 0, 230, 70, 40, 50, 40, 12, 0.8],
       [0.02, 4.5, 70, 1, 240, 85, 40, 75, 68, 10, 0.8],
       [0.05, 6.0, 10, 1, 245, 75, 85, 75, 205, 5, 0.5],
       [0.15, 1.0, 50, 0, 220, 60, 120, 0, 120, 16, 1.5]], dtype=object)
```

```python
# TRAIN 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=0.2,random_state=0)
```

```python
# FEATURE SCALING

from sklearn.preprocessing import MinMaxScaler

sc=MinMaxScaler()
```

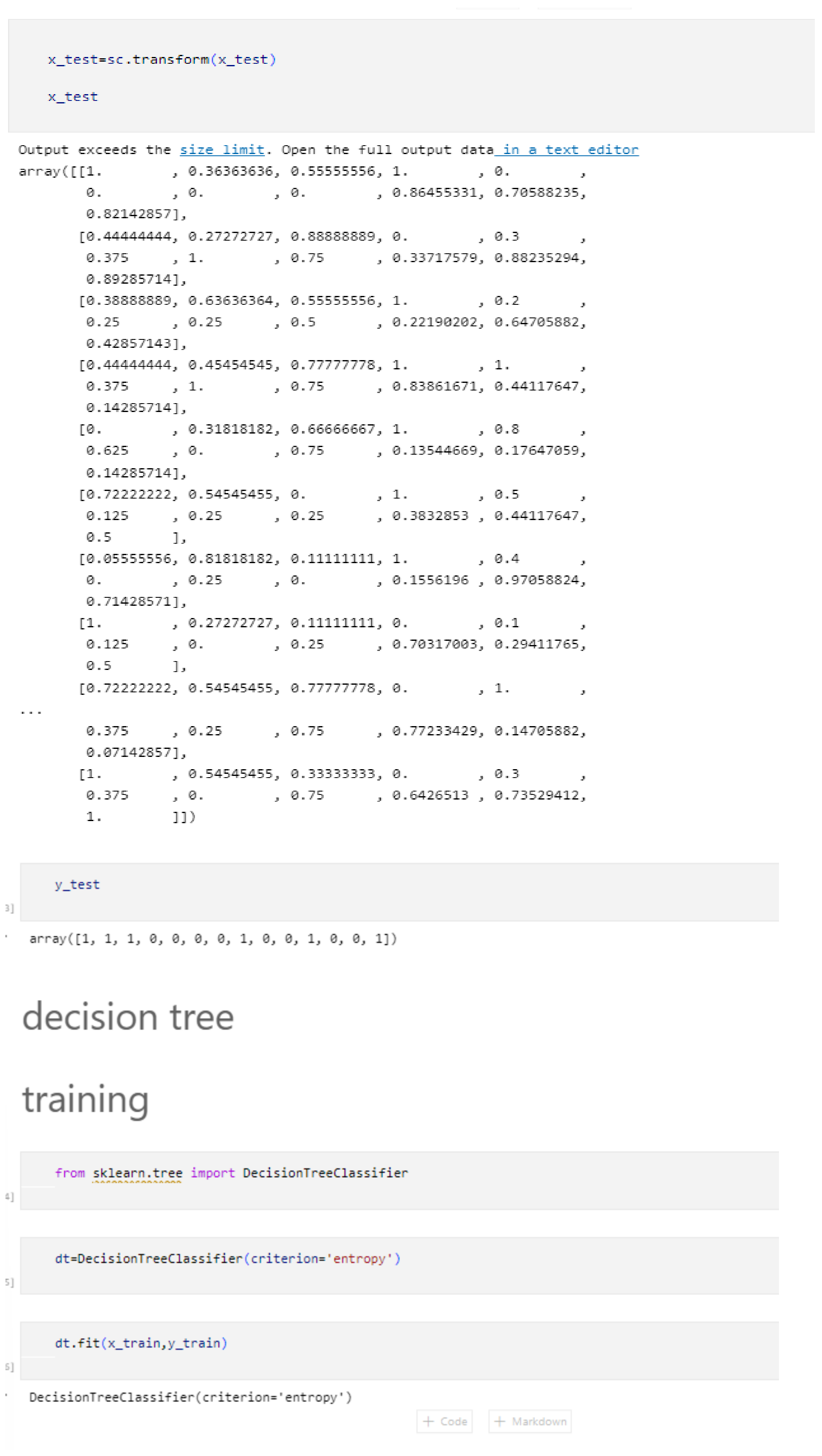
```
x_train
```

```
Output exceeds the size limit. Open the full output data in a text editor
array([[0.15, 1.0, 50, 0, 220, 60, 120, 0, 120, 16, 1.5],
       [0.02, 1.0, 80, 0, 230, 70, 40, 50, 40, 8, 0.8],
       [0.06, 2.0, 20, 1, 225, 65, 60, 25, 92, 12, 1.4],
       [0.15, 4.0, 50, 0, 220, 60, 60, 0, 168, 27, 2.4],
       [0.06, 6.0, 80, 0, 220, 60, 60, 0, 75, 37, 2.4],
       [0.1, 1.0, 50, 0, 230, 70, 120, 50, 265, 10, 0.9],
       [0.2, 9.0, 90, 1, 225, 65, 40, 25, 276, 34, 3.1],
       [0.05, 6.0, 10, 1, 245, 75, 85, 75, 205, 5, 0.5],
       [0.15, 6.0, 50, 0, 230, 70, 60, 50, 225, 18, 1.4],
       [0.02, 10.0, 10, 1, 210, 70, 40, 50, 21, 14, 1.5],
       [0.06, 3.0, 50, 0, 250, 80, 60, 100, 220, 10, 1.0],
       [0.1, 4.0, 40, 0, 205, 65, 120, 25, 176, 12, 1.2],
       [0.1, 3.0, 50, 1, 210, 70, 120, 50, 128, 18, 1.8],
       [0.1, 4.0, 95, 0, 220, 75, 120, 100, 121, 14, 1.5],
       [0.2, 7.0, 30, 0, 230, 70, 40, 50, 298, 28, 2.2],
       [0.2, 6.0, 90, 1, 240, 75, 40, 75, 360, 28, 1.6],
       [0.06, 12.0, 50, 1, 230, 80, 65, 100, 74, 29, 2.1],
       [0.06, 5.0, 90, 0, 215, 95, 60, 75, 92, 38, 2.2],
       [0.04, 2.0, 80, 0, 230, 70, 40, 50, 40, 12, 0.8],
       [0.06, 10.0, 90, 1, 200, 60, 60, 0, 126, 27, 2.2],
       [0.02, 10.0, 40, 1, 200, 60, 40, 0, 60, 24, 1.1],
       [0.06, 3.0, 40, 0, 205, 65, 60, 25, 145, 23, 1.9],
       [0.1, 1.0, 40, 0, 220, 60, 120, 0, 120, 16, 1.2],
       [0.15, 3.0, 10, 1, 225, 65, 70, 25, 154, 19, 1.8],
       [0.02, 9.0, 70, 0, 215, 75, 40, 75, 24, 27, 1.4],
...
       [0.02, 4.0, 70, 1, 240, 75, 40, 75, 68, 10, 0.5],
       [0.02, 8.0, 90, 0, 220, 60, 40, 0, 25, 18, 1.2],
       [0.06, 10.0, 100, 1, 200, 60, 60, 0, 126, 27, 2.2],
       [0.2, 5.0, 60, 1, 210, 70, 40, 50, 278, 30, 3.2],
       [0.2, 3.0, 80, 0, 250, 80, 40, 100, 357, 21, 1.1]], dtype=object)
```

```
x_train=sc.fit_transform(x_train)
```

```
x_train
```

```
Output exceeds the size limit. Open the full output data in a text editor
array([[0.72222222, 0.        , 0.44444444, 0.        , 0.4       ,
        0.        , 1.        , 0.        , 0.28530259, 0.35294118,
        0.39285714],
       [0.        , 0.        , 0.77777778, 0.        , 0.6       ,
        0.25      , 0.        , 0.5       , 0.05475504, 0.11764706,
        0.14285714],
       [0.22222222, 0.09090909, 0.11111111, 1.        , 0.5       ,
        0.125     , 0.25      , 0.25      , 0.20461095, 0.23529412,
        0.35714286],
       [0.72222222, 0.27272727, 0.44444444, 0.        , 0.4       ,
        0.        , 0.25      , 0.        , 0.42363112, 0.67647059,
        0.71428571],
       [0.22222222, 0.45454545, 0.77777778, 0.        , 0.4       ,
        0.        , 0.25      , 0.        , 0.1556196 , 0.97058824,
        0.71428571],
       [0.44444444, 0.        , 0.44444444, 0.        , 0.6       ,
        0.25      , 1.        , 0.5       , 0.70317003, 0.17647059,
        0.17857143],
       [1.        , 0.72727273, 0.88888889, 1.        , 0.5       ,
        0.125     , 0.        , 0.25      , 0.73487032, 0.88235294,
        0.96428571],
       [0.16666667, 0.45454545, 0.        , 1.        , 0.9       ,
        0.375     , 0.5625    , 0.75      , 0.53025937, 0.02941176,
        0.03571429],
       [0.72222222, 0.45454545, 0.44444444, 0.        , 0.6       ,
...
        0.25      , 0.        , 0.5       , 0.74063401, 0.76470588,
        1.        ],
       [1.        , 0.18181818, 0.77777778, 0.        , 1.        ,
        0.5       , 0.        , 1.        , 0.96829971, 0.5       ,
        0.25      ]])
```



```
x_test=sc.transform(x_test)

x_test
```

```
Output exceeds the size limit. Open the full output data in a text editor
array([[1.        , 0.36363636, 0.55555556, 1.        , 0.        ,
        0.        , 0.        , 0.        , 0.86455331, 0.70588235,
        0.82142857],
       [0.44444444, 0.27272727, 0.88888889, 0.        , 0.3       ,
        0.375     , 1.        , 0.75      , 0.33717579, 0.88235294,
        0.89285714],
       [0.38888889, 0.63636364, 0.55555556, 1.        , 0.2       ,
        0.25      , 0.25      , 0.5       , 0.22190202, 0.64705882,
        0.42857143],
       [0.44444444, 0.45454545, 0.77777778, 1.        , 1.        ,
        0.375     , 1.        , 0.75      , 0.83861671, 0.44117647,
        0.14285714],
       [0.        , 0.31818182, 0.66666667, 1.        , 0.8       ,
        0.625     , 0.        , 0.75      , 0.13544669, 0.17647059,
        0.14285714],
       [0.72222222, 0.54545455, 0.        , 1.        , 0.5       ,
        0.125     , 0.25      , 0.25      , 0.3832853 , 0.44117647,
        0.5       ],
       [0.05555556, 0.81818182, 0.11111111, 1.        , 0.4       ,
        0.        , 0.25      , 0.        , 0.1556196 , 0.97058824,
        0.71428571],
       [1.        , 0.27272727, 0.11111111, 0.        , 0.1       ,
        0.125     , 0.        , 0.25      , 0.70317003, 0.29411765,
        0.5       ],
       [0.72222222, 0.54545455, 0.77777778, 0.        , 1.        ,
...
        0.375     , 0.25      , 0.75      , 0.77233429, 0.14705882,
        0.07142857],
       [1.        , 0.54545455, 0.33333333, 0.        , 0.3       ,
        0.375     , 0.        , 0.75      , 0.6426513 , 0.73529412,
        1.        ]])
```

```
y_test
```

```
array([1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1])
```

decision tree

training

```
from sklearn.tree import DecisionTreeClassifier
```

```
dt=DecisionTreeClassifier(criterion='entropy')
```

```
dt.fit(x_train,y_train)
```

```
DecisionTreeClassifier(criterion='entropy')
```

predicting

```
y_pred_dt=dt.predict(x_test)
```

```
y_pred_dt
```

```
array([1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1])
```

```
import sklearn.metrics as metrics
```

```
fpr,tpr,threshold=metrics.roc_curve(y_test,y_pred_dt)
```

```
roc_auc_DT=metrics.auc(fpr,tpr)
```

```
roc_auc_DT
```

```
0.9375
```

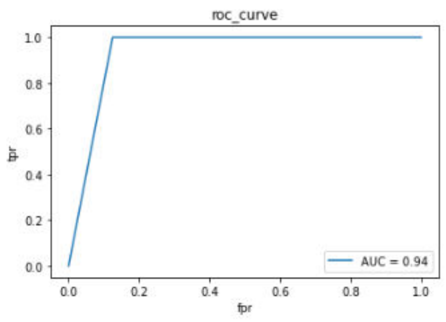
```
from sklearn.metrics import accuracy_score
```

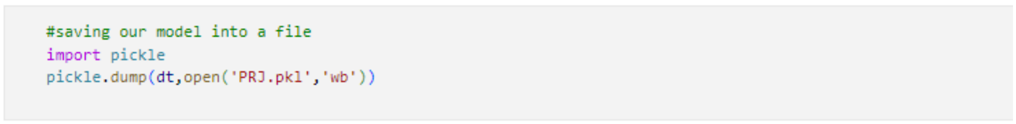
```
accuracy_score(y_test,y_pred_dt)
```

```
0.9285714285714286
```

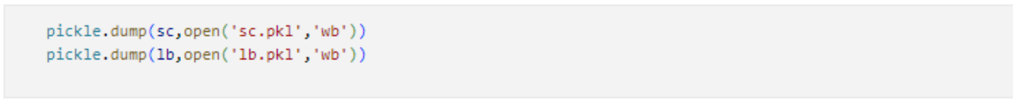
```
plt.plot(fpr,tpr,label='AUC = %0.2f' % roc_auc_DT)
plt.xlabel("fpr")
plt.ylabel("tpr")
plt.title("roc_curve")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x27b31949f70>
```





```
#saving our model into a file
import pickle
pickle.dump(dt,open('PRJ.pkl','wb'))
```

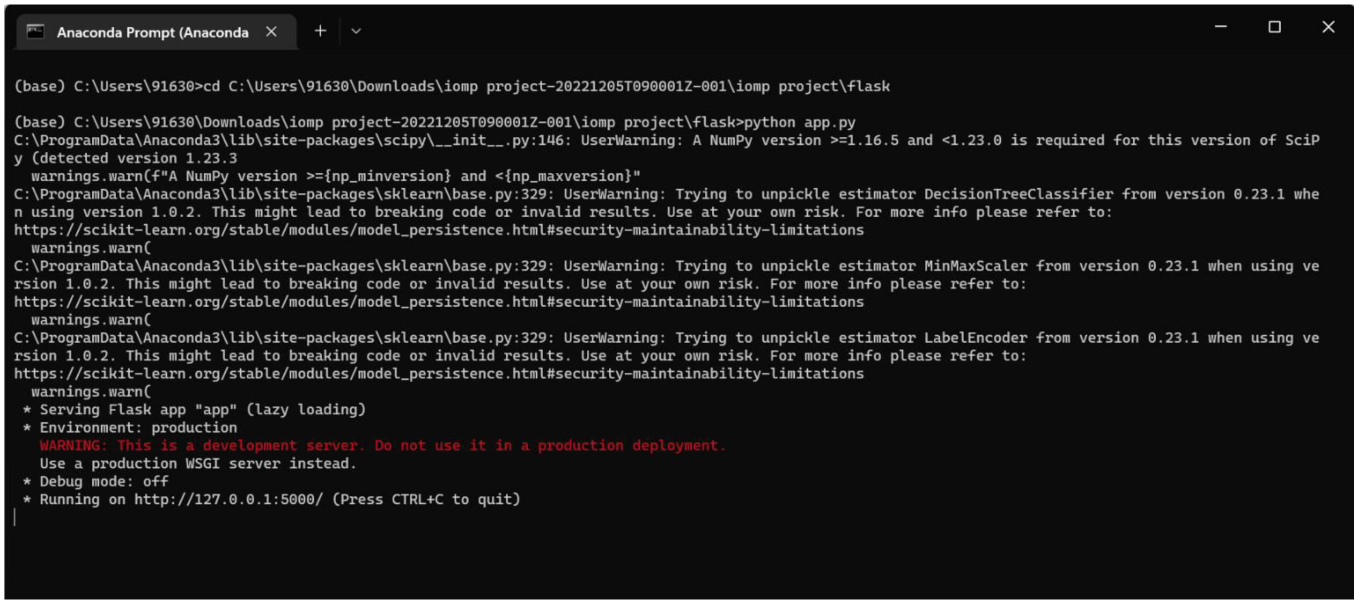


```
pickle.dump(sc,open('sc.pkl','wb'))
pickle.dump(lb,open('lb.pkl','wb'))
```

App.py

```
# -*- coding: utf-8 -*-
"""
Created on Thu Feb  4 16:43:40 2021
@author: supriya
"""
import numpy as np
import pandas as pd
import pickle
import random
from flask import Flask,request, render_template

app=Flask(__name__,template_folder="templates")
model = pickle.load(open('PRJ.pkl', 'rb'))
sc = pickle.load(open('sc.pkl', 'rb'))
lb = pickle.load(open('lb.pkl', 'rb'))

@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def about():
    return render_template('home.html')
@app.route('/pred',methods=['GET'])
def page():
    return render_template('result.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]
    features_name =
['layer_height','wall_thickness','infill_density','infill_pattern','nozzle_temperature'
,
'bed_temperature','print_speed','fan_speed','roughness','tension_strenght','elongation'
]
```

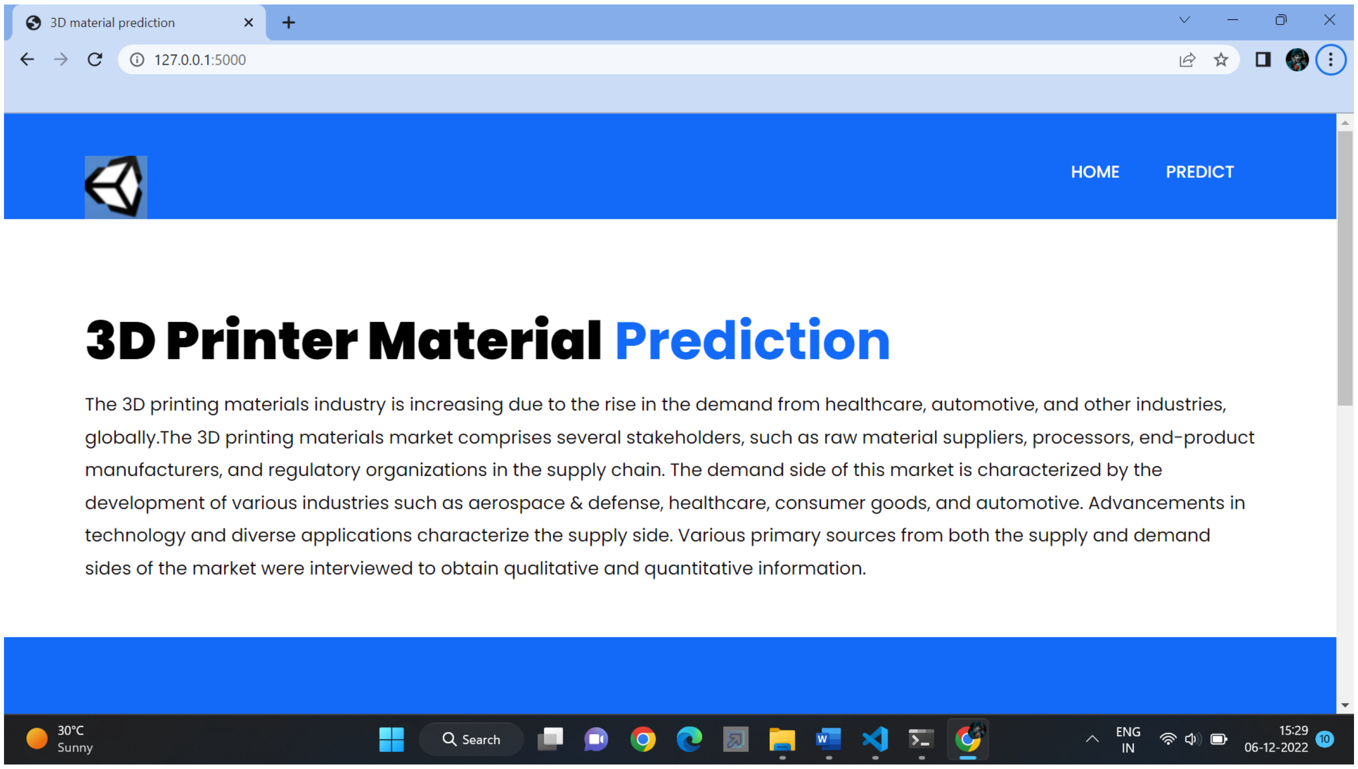
```
    x_df=pd.DataFrame(features_value)
    a=random.randint(0,1)
    print(a)




    output=a
    print(output)
    if(output==1) :
        return render_template("result.html",prediction_text = "The Suggested Material
is ABS.(Acrylonitrile butadiene styrene is a common thermoplastic polymer typically
used for injection molding applications)")
    elif(output==0) :
        return render_template("result.html",prediction_text = "The Suggested Material
is PLA.(PLA, also known as polylactic acid or polylactide, is a thermoplastic made from
renewable resources such as corn starch, tapioca roots or sugar cane, unlike other
industrial materials made primarily from petroleum)")
    else :
        return render_template("result.html",prediction_text = 'The given values do not
match the range of values of the model.Try giving the values in the mnetioned range')


if __name__ == '__main__':
     app.run( debug=False)
```



Anaconda Prompt (Anaconda

```
(base) C:\Users\91630>cd C:\Users\91630\Downloads\iomp project-20221205T090001Z-001\iomp project\flask

(base) C:\Users\91630\Downloads\iomp project-20221205T090001Z-001\iomp project\flask>python app.py
C:\ProgramData\Anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciP
y (detected version 1.23.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 0.23.1 whe
n using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator MinMaxScaler from version 0.23.1 when using ve
rsion 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator LabelEncoder from version 0.23.1 when using ve
rsion 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  warnings.warn(
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

11.RESULTS



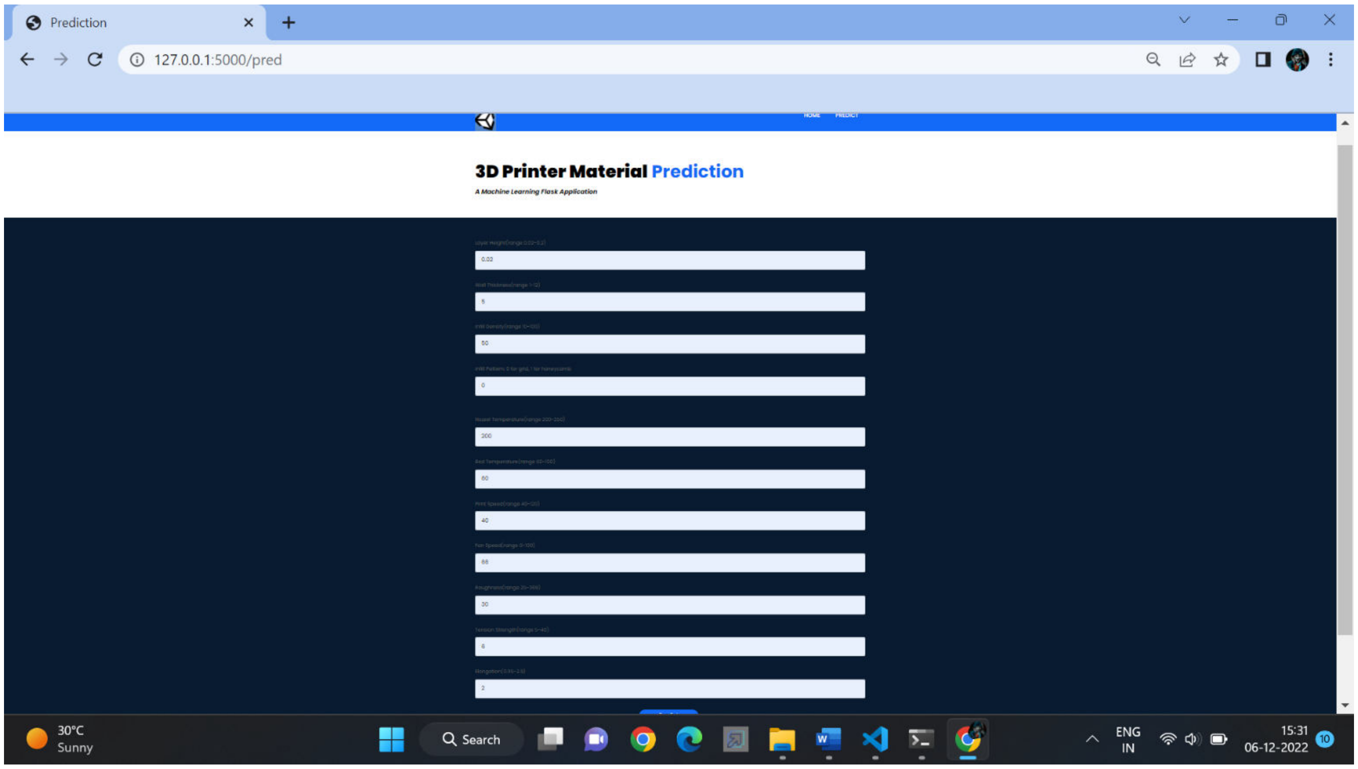
3D material prediction

127.0.0.1:5000

HOME PREDICT

3D Printer Material Prediction

The 3D printing materials industry is increasing due to the rise in the demand from healthcare, automotive, and other industries, globally.The 3D printing materials market comprises several stakeholders, such as raw material suppliers, processors, end-product manufacturers, and regulatory organizations in the supply chain. The demand side of this market is characterized by the development of various industries such as aerospace & defense, healthcare, consumer goods, and automotive. Advancements in technology and diverse applications characterize the supply side. Various primary sources from both the supply and demand sides of the market were interviewed to obtain qualitative and quantitative information.



Prediction

127.0.0.1:5000/pred

3D Printer Material Prediction

A Machine Learning Flask Application

Layer Height(range 0.02-0.2)
0.02

Wall Thickness(range 1-12)
5

Infill Density(range 10-100)
50

Infill Pattern: 0 for grid, 1 for honeycomb
0

Nozzel Temperature(range 200-250)
200

Bed Temperature(range 60-100)
60

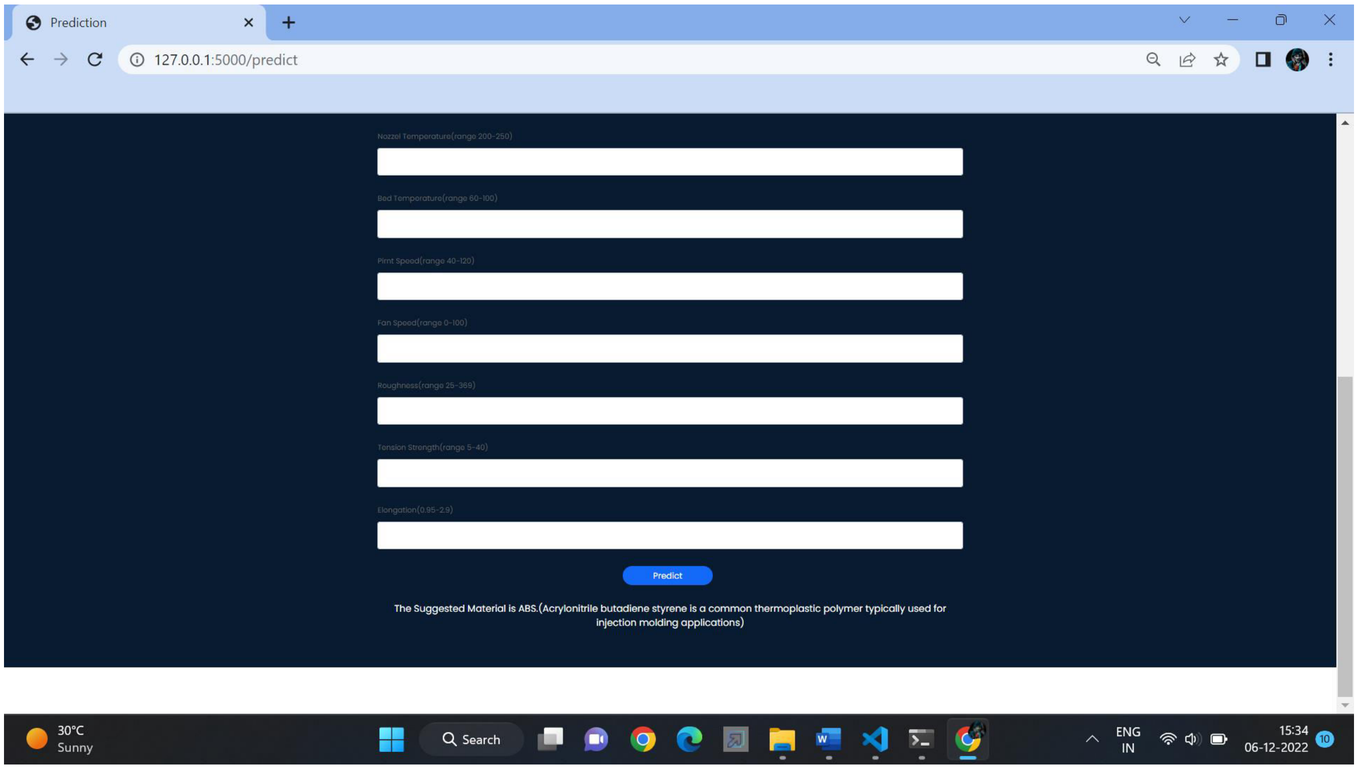
Print Speed(range 40-120)
40

Fan Speed(range 0-100)
66

Roughness(range 20-368)
30

Tension Strength(range 5-40)
6

Elongation(0.95-2.8)
2



Nozzel Temperature(range 200-250)

Bed Temperature(range 60-100)

Pirnt Speed(range 40-120)

Fan Speed(range 0-100)

Roughness(range 25-369)

Tension Strength(range 5-40)

Elongation(0.95-2.9)

Predict

The Suggested Material is ABS.(Acrylonitrile butadiene styrene is a common thermoplastic polymer typically used for injection molding applications)