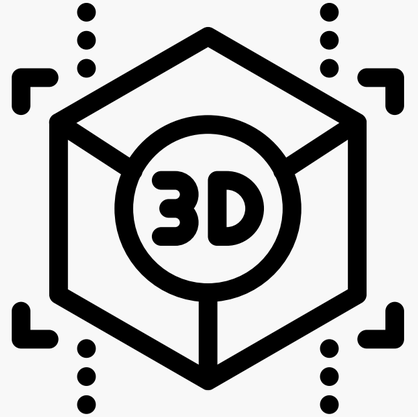
3D PRINTER MONITORING SYSTEM

BY: DILJOT SINGH AND FRANCESCA MISON (HAMK, FINLAND)

ENGINEERING APPLICATIONS IN PYTHON - SAP
TUS LIMERICK, JUNE 24 - 28, 2024

OVERVIEW:



What is this?:

- Python-powered 3D printer monitoring system

What does it do?

- Tracks real-time data parameters, provides actionable insights, and minimizes downtime through predictive maintenance

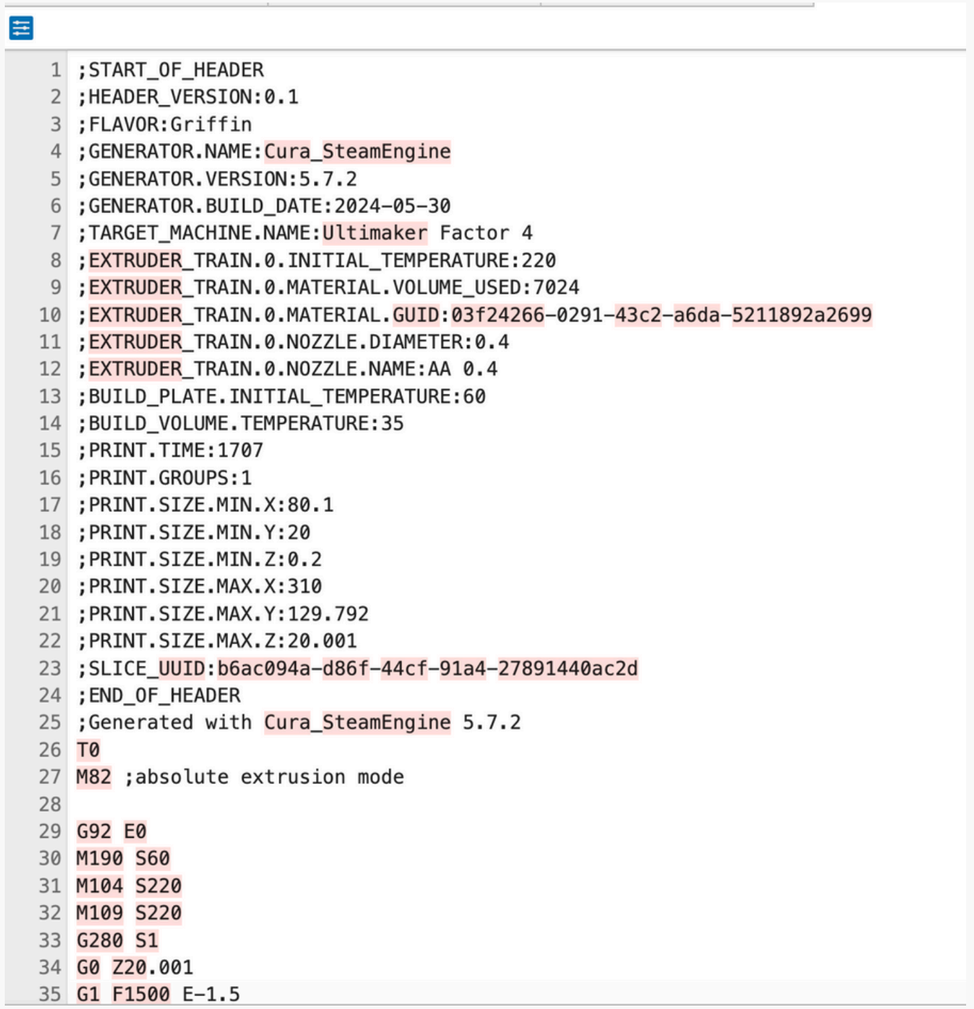
Why should we use a system like this?

- Operational oversight (real-time monitoring) and enhanced proactive maintenance (timely interventions to prevent issues).

COLLABORATION

This project was done in collaboration with the group that is focusing on 3D Printing Path Generator.

The G-Code produced was provided, then we used that to analyze and visualize the data.

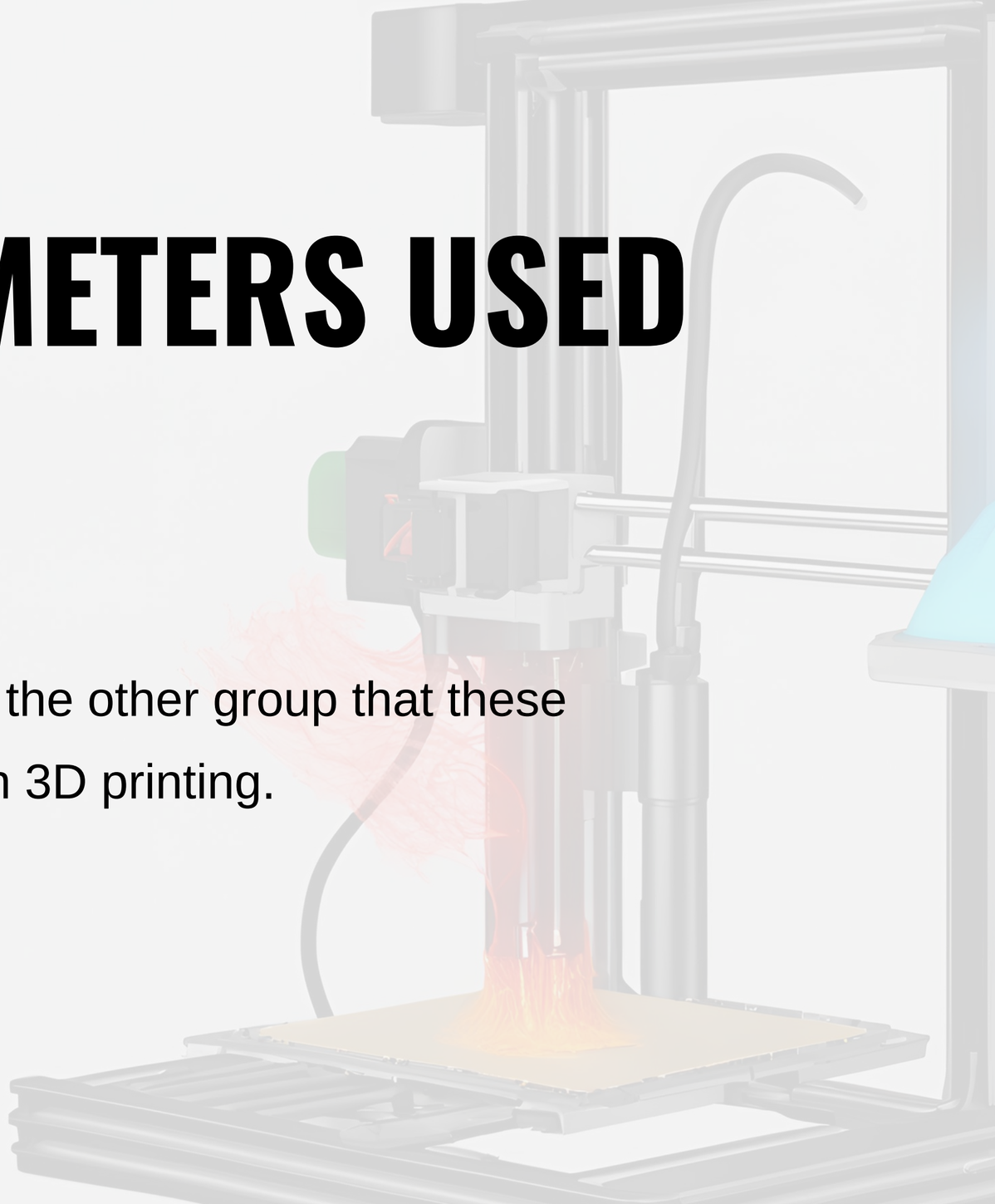


```
;START_OF_HEADER
;HEADER_VERSION:0.1
;FLAVOR:Griffin
;GENERATOR.NAME:Cura_SteamEngine
;GENERATOR.VERSION:5.7.2
;GENERATOR.BUILD_DATE:2024-05-30
;TARGET_MACHINE.NAME:Ultimaker Factor 4
;EXTRUDER_TRAIN.0.INITIAL_TEMPERATURE:220
;EXTRUDER_TRAIN.0.MATERIAL.VOLUME_USED:7024
;EXTRUDER_TRAIN.0.MATERIAL.GUID:03f24266-0291-43c2-a6da-5211892a2699
;EXTRUDER_TRAIN.0.NOZZLE.DIAMETER:0.4
;EXTRUDER_TRAIN.0.NOZZLE.NAME:AA 0.4
;BUILD_PLATE.INITIAL_TEMPERATURE:60
;BUILD_VOLUME.TEMPERATURE:35
;PRINT.TIME:1707
;PRINT.GROUPS:1
;PRINT.SIZE.MIN.X:80.1
;PRINT.SIZE.MIN.Y:20
;PRINT.SIZE.MIN.Z:0.2
;PRINT.SIZE.MAX.X:310
;PRINT.SIZE.MAX.Y:129.792
;PRINT.SIZE.MAX.Z:20.001
;SLICE_UUID:b6ac094a-d86f-44cf-91a4-27891440ac2d
;END_OF_HEADER
;Generated with Cura_SteamEngine 5.7.2
T0
M82 ;absolute extrusion mode

G92 E0
M190 S60
M104 S220
M109 S220
G280 S1
G0 Z20.001
G1 F1500 E-1.5
```

VARIABLES/PARAMETERS USED

Nozzle, Plate, and Volume Temperature.

We used these because we were advised from the other group that these are the most important factors to consider when 3D printing.



LIBRARIES USED

```
#Libraries
```

```
import pandas as pd
import numpy as np
import time
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
```

- **Pandas (‘pd’)**: for data manipulation and data input/output
- **Numpy (‘np’)**: for numerical computation and random number generation
- **Time (‘time’)**: time management
- **Matplotlib (‘plt’)**: data visualization
- **Seaborn (‘sns’)**: statistical data visualization
- **Scikit-Learn (‘train_test_split, RandomForestClassifier, classification_report, accuracy_score’)**: model selection, machine learning models, and model evaluation

RANGES

These ranges are based on typical 3D printer specifications to ensure operations within optimal and safe conditions.

- **temperature_extruder**: 200°C to 240°C
- **temperature_bed**: 50°C to 70°C
- **vibrations**: 0 to 1 (arbitrary units, assuming lower values are better)
- **print_time**: 1200 seconds to 1600 seconds
- **material_volume_used**: 5000 units to 5500 units

DATA COLLECTION

FUNCTIONS USED TO SIMULATE

- **get_temperature_extruder()**: Simulates extruder temperature around 220°C.
- **get_temperature_bed()**: Simulates bed temperature around 60°C.
- **get_vibrations()**: Simulates vibrations, taking absolute values to avoid negative numbers.
- **get_print_time()**: Simulates print time between 1200 and 1600 seconds.
- **get_material_volume_used()**: Simulates material volume used between 5000 and 5500 units.

MONITORING FUNCTION

Checks whether or not the values in each parameter are within range, and alerts a message if it does not.

```
def check_values(data_row):
    alerts = []
    for param, (min_val, max_val) in acceptable_ranges.items():
        if not (min_val <= data_row[param] <= max_val):
            alerts.append(f"Alert: {param} out of range. Current value: {data_row[param]}")
    return alerts
```

Function for checking values

Loop: (‘for’)

Condition: (‘if’)

List operations: (‘alerts = [ ]’) & (‘alerts.append’)

DATA COLLECTION AND MONITORING LOOP

This loop simulates the real-time data collection process. It collects data, checks the values, and prints alerts if any parameter is out of range. The collected data is stored in a DataFrame and saved to a CSV file.

```
columns = ['timestamp', 'temperature_extruder', 'temperature_bed', 'volume_temperature']
data = pd.DataFrame(columns=columns)

try:
    for _ in range(100):
        data_row = {
            'timestamp': time.time(),
            'temperature_extruder': get_temperature_extruder(),
            'temperature_bed': get_temperature_bed(),
            'volume_temperature': get_volume_temperature()
        }
        data_row_df = pd.DataFrame([data_row])
        if not data_row_df.isnull().all(axis=None):
            data = pd.concat([data, data_row_df], ignore_index=True)
        alerts = check_values(data_row)
        if alerts:
            for alert in alerts:
                print(alert)
            print("Stopping the 3D printer due to the above issues.")
            break
        print(data_row)
        time.sleep(0.5)
except KeyboardInterrupt:
    print("Data collection stopped manually.")


data.to_csv('3d_printer_data_simulated.csv', index=False)
print("Data collection completed and saved to '3d_printer_data_simulated.csv'")
```

Python Commands Used:

- **DataFrame Creation: pd.DataFrame(), pd.concat(), data.to_csv()**
- **List: [ ] for defining column names.**
- **For Loop: for _ in range(100): for iterative data collection.**
- **Function Calls: get_temperature_extruder(), get_temperature_bed(), get_volume_temperature(), check_values()**
- **Dictionary: {} for creating data_row.**
- **Conditional: if, not, isnull().all()**
- **Try-Except: try, except KeyboardInterrupt:**
- **Print: print()**
- **Time Delay: time.sleep(0.5)**

VISUALIZATION

To visualize the collected data, the following code plots the variations of different parameters over time.

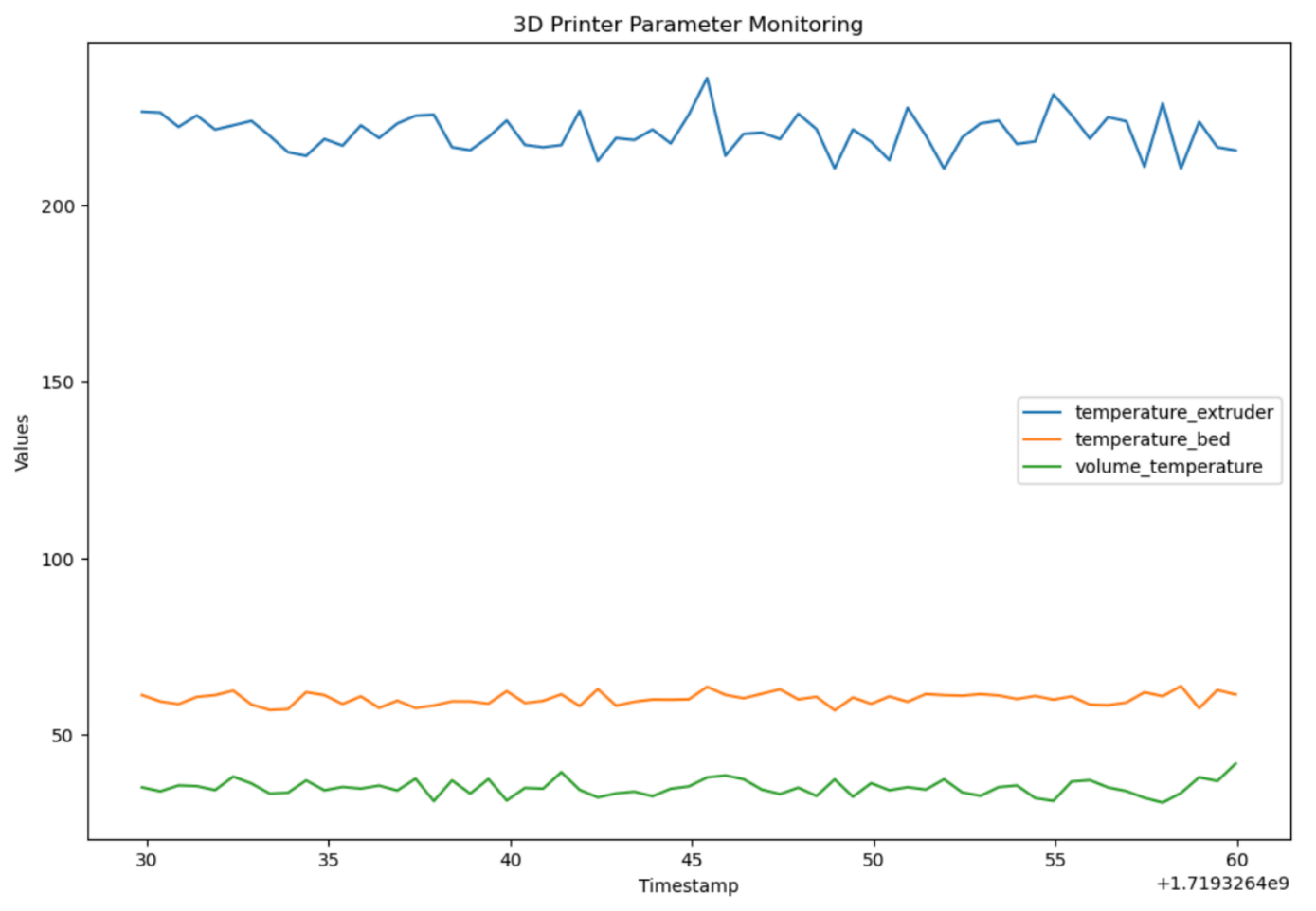
PARAMETERS:

```
df = pd.read_csv('3d_printer_data_simulated.csv')


df.head()

plt.figure(figsize=(12, 8))
for column in columns[1:]:  # Exclude 'timestamp'
    plt.plot(df['timestamp'], df[column], label=column)
plt.xlabel('Timestamp')
plt.ylabel('Values')
plt.title('3D Printer Parameter Monitoring')
plt.legend()
plt.show()
```

Python Commands Used:

- **Data Loading: pd.read_csv()**
- **Missing Value Handling: df.fillna()**
- **Column Dropping: df.drop()**
- **Data Scaling: MinMaxScaler().fit_transform()**
- **DataFrame Creation: pd.DataFrame()**
- **Random Data Generation: np.random.choice()**
- **Display Data: head()**

VISUALIZATION



Graph that visualizes time series data of various 3D printer parameters over time, used for monitoring and identifying performance trends and potential issues.

VISUALIZATION

To visualize the collected data, the following code plots the variations of different parameters over time.

GRAPHS:

```
# Plot feature importance
feature_importance = model.feature_importances_
features_names = features.columns

# Create a DataFrame for visualization
feature_importance_df = pd.DataFrame({
    'Feature': features_names,
    'Importance': feature_importance
}).sort_values(by='Importance', ascending=False)

# Plot
plt.figure(figsize=(10, 6))
sns.barplot(data=feature_importance_df, x='Importance', y='Feature')
plt.title('Feature Importance for Predicting Maintenance Needs')
plt.show()
```

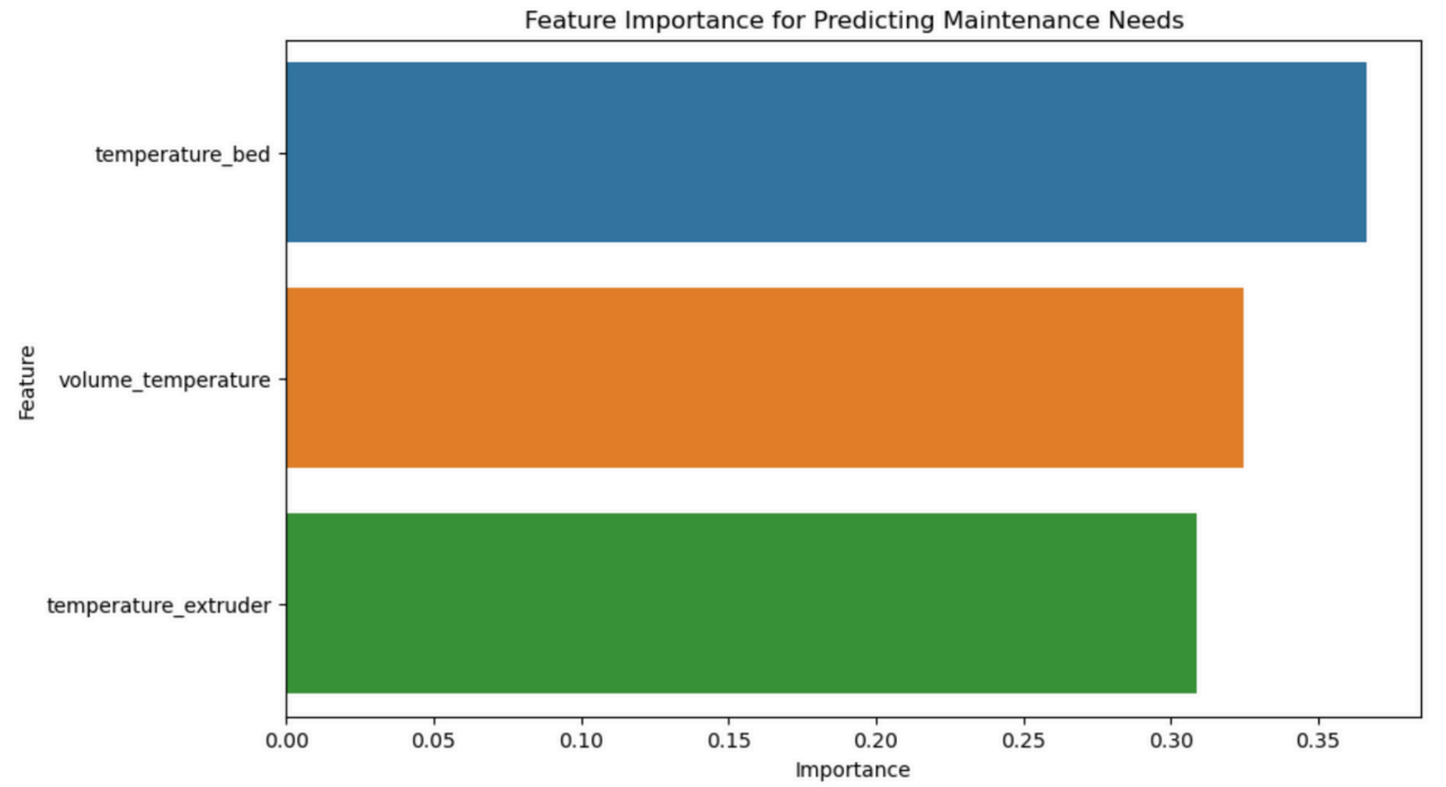
Python Commands Used:

- **Accessing Attributes: model.feature_importances_ retrieves the feature importance array from the model.**
- **Getting Column Names: features.columns accesses the column names of the DataFrame.**
- **Creating DataFrame: pd.DataFrame({...}) creates a DataFrame from the dictionary, and sort_values() sorts it.**

Plotting:

- **plt.figure(figsize=(...)) initializes the plot with a specified size.**
- **sns.barplot(data=..., x='...', y='...') creates a bar plot with specified data and axes.**
- **plt.title('...') adds a title to the plot.**
- **plt.show() displays the plot.**

VISUALIZATION



Graph plotted into DataFrame to highlight the feature importance to predict which parameters need maintenance.

RESULTS

```
Alert: volume_temperature out of range. Current value: 41.618558344130776
Stopping the 3D printer due to the above issues.
Data collection completed and saved to '3d_printer_data_simulated.csv'
```

- Creates an alert if any of the parameters go out of range
- Stops the 3D printing process based on the alerts
- Collects data to analyze for further use which will improve the system

THANK YOU! (KIITOS)