Problem Description

In the context of 3D printing, the selection of printing material significantly influences the characteristics of the final printed object. This project aims to create a predictive model that can accurately determine the material type used in 3D printing based on the provided configuration settings. The configuration settings encompass parameters like layer height, wall thickness, infill density, infill pattern, nozzle temperature, bed temperature, print speed, fan speed, and roughness. The goal is to build a classification model that can categorize each configuration as either "ABS" (Acrylonitrile Butadiene Styrene) or "PLA" (Polylactic Acid), which are two common 3D printing materials. This model will enable efficient material selection during the 3D printing process, contributing to improved manufacturing processes and higher-quality printed objects.

Dataset Description

This dataset comes from research by TR/Selcuk University Mechanical Engineering department.

Input Parameters:

- ✔ layer_height: The thickness of each layer in the printed object, measured in millimeters (mm).
- ✔ wall_thickness: The thickness of the walls (outer layers) of the printed object, measured in millimeters (mm).
- ✔ infill_density: The density of the infill, which is the interior structure of the printed object. It's usually represented as a percentage.
- ✔ infill_pattern: The pattern used for the infill, which can affect the object's strength and material usage. Common patterns are "grid" and "honeycomb."
- ✔ nozzle_temperature: The temperature of the 3D printer's nozzle, which melts the filament material for deposition, measured in degrees Celsius (°C).
- ✔ bed_temperature: The temperature of the printer's build surface (bed), which helps with adhesion, measured in degrees Celsius (°C).
- ✔ print_speed: The speed at which the printer's nozzle moves while printing, measured in millimeters per second (mm/s).
- ✔ fan_speed: The speed of any cooling fans that are directed at the printed object, measured as a percentage.
- ✔ roughness: A measure of the surface texture or smoothness of the printed object. The values provided are not labeled, but they seem to indicate some kind of roughness measurement.
- ✔ tension_strength: The maximum amount of stress the printed object can withstand without breaking, measured in some unspecified units.

- ✔ elongation: The amount a material can stretch without permanently deforming, usually expressed as a percentage.

Output Parameter:

- ✔ material: The type of material being used for printing, in this case, ABS (Acrylonitrile Butadiene Styrene).

Choice of Algorithm

In the pursuit of identifying the most suitable classification algorithm for the task of predicting 3D printing material based on configuration settings, a comprehensive evaluation of multiple algorithms was performed. The tested algorithms encompassed Logistic Regression, Decision Tree, Support Vector Machine (SVM), Random Forest, Gradient Boosting, k-Nearest Neighbors (kNN), Gaussian Naive Bayes (Gaussian NB), MLP Classifier (Multi-Layer Perceptron), and Linear Discriminant Analysis (LDA).

Through rigorous cross-validation utilizing the StratifiedKFold technique, where class imbalances were taken into account, each algorithm's performance was scrutinized across 5 folds. After careful assessment, the MLP Classifier emerged as the optimal choice. The MLP Classifier demonstrated a balance between complexity and expressiveness, resulting in favorable predictive performance for the given task.

The final selection of the MLP Classifier was based on its capacity to capture intricate relationships within the dataset, as well as its adaptability to a range of problem domains. Its inclusion in this predictive model aims to effectively classify 3D printing materials, thus contributing to efficient material selection and enhancing the quality of printed objects.

Description of key steps

- ■ **Label Encoding**

Label encoding is a method used to convert categorical data into numerical form. It's particularly useful when dealing with ordinal categorical variables, where there's a meaningful order among the categories. Each category is assigned a unique numerical label, allowing machine learning algorithms to work with this data.

- **Data Splitting**

Data splitting is a crucial step in machine learning. It involves dividing a dataset into two parts: training data, used to teach the model, and testing data, kept unseen to evaluate the model's performance on new information.

- Training Data: This part trains the model. It contains input features and corresponding target labels. The model learns patterns and relationships from this data.

- Testing Data: This part assesses the model's performance. It's unseen during training and contains only input features. The model's predictions are compared with true labels to measure its effectiveness.

Common methods include:

Train-Test Split: Divides data into training and testing sets (e.g., 80-20%). The model learns from the training set and is tested on the separate test set.

Data splitting helps prevent overfitting and gauges how well the model generalizes to new data.

- **Data Normalization**

Data normalization is the process of transforming numerical data to a common scale. It ensures that features have similar magnitudes, preventing certain variables from dominating others. This aids machine learning algorithms in converging faster and being less sensitive to the scale of input features.

- **Cross Validation**

Cross-validation is a technique used to assess the performance of machine learning models. It involves dividing the dataset into multiple subsets or "folds." The model is trained on a subset and tested on the remaining data. This process is repeated, with each fold serving as both training and test data. Cross-validation provides a more robust evaluation of a model's generalization ability and helps detect overfitting.

- **Feature Engineering**

Feature engineering is the process of selecting, transforming, or creating new features from raw data to enhance a machine learning model's performance. It involves extracting relevant information, reducing noise, and improving the representation of data. Effective feature engineering can significantly impact a model's accuracy and predictive power, ultimately leading to better outcomes.

- **Hyperparameter Tuning**

Hyperparameter tuning involves fine-tuning the parameters of a machine learning algorithm that are not learned during training (unlike model parameters). These hyperparameters influence how the model learns and generalizes. By experimenting with different values, data scientists aim to optimize the model's performance. Techniques like grid search and random search are commonly used to systematically explore the hyperparameter space and find the best combination for optimal results.

- **Model Building**

Model building is the core of machine learning, where data is used to train algorithms to make predictions or classifications. It involves selecting an appropriate algorithm, preparing data, and tuning hyperparameters. Model building aims to create a reliable representation of relationships within the data. Successful model building requires a deep understanding of algorithms, feature engineering, and performance evaluation to achieve accurate and meaningful predictions.

Obtained Results

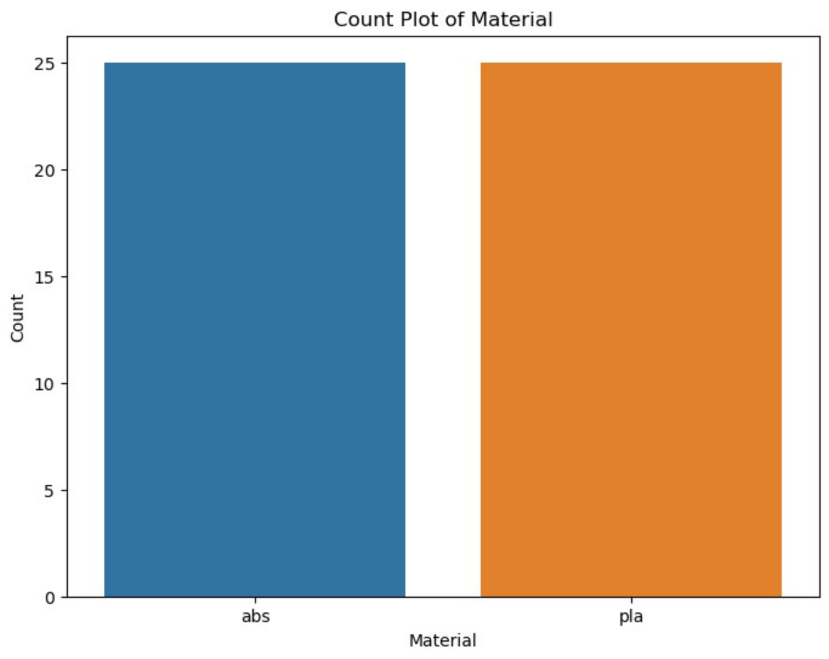

Fig-1: Count of Target Labels

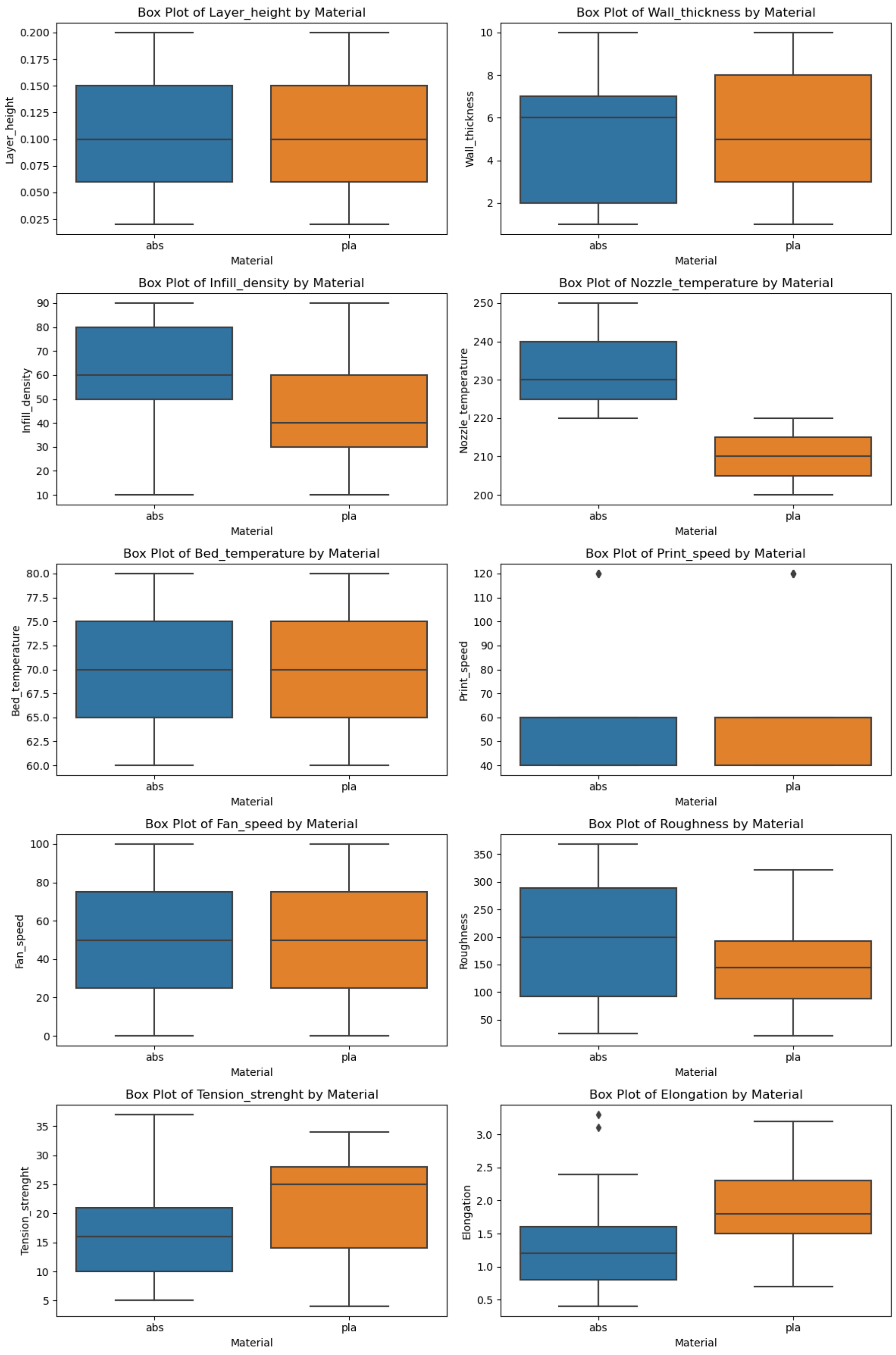

Fig-2: Other features boxplot against material column

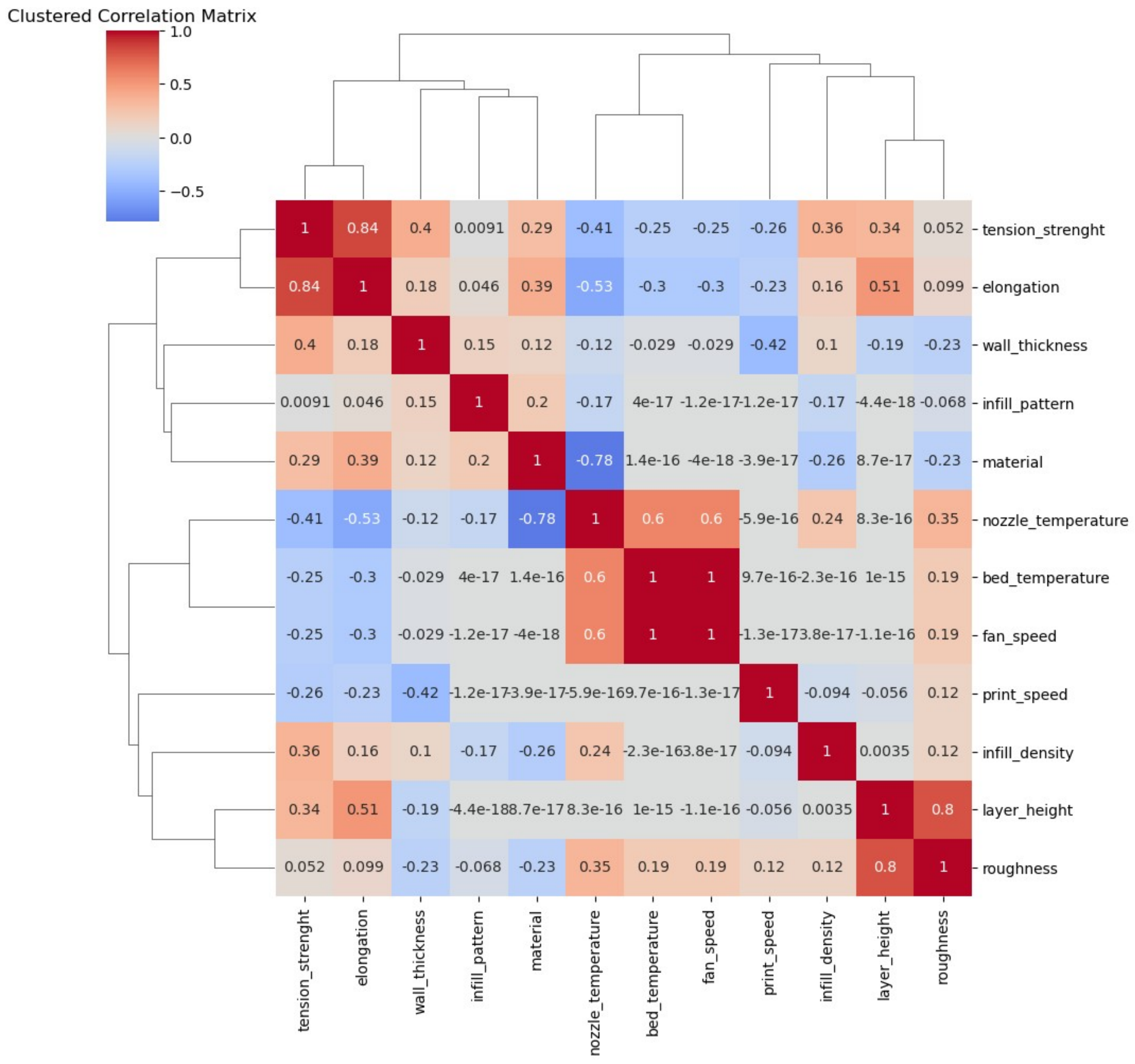

Fig-3: Cluster Map of Correlation Matrix

Table-1: Cross-validation Model Performance

Model Name	**AUC**	**Specificity**	**Sensitivity**	**Accuracy**	**Precision**	**F1Score**	**Recall**
Logistic Regression	1	1	1	1	1	1	1
Decision Tree	0.7879	0.9091	0.6667	0.8	0.8571	0.75	0.6667
SVM	0.8813	0.8182	0.9444	0.875	0.8095	0.8718	0.9444
Random Forest	0.8939	0.9545	0.8333	0.9	0.9375	0.8824	0.8333
Gradient Boosting	0.7879	0.9091	0.6667	0.8	0.8571	0.75	0.6667
kNN	0.7854	0.6818	0.8889	0.775	0.6957	0.7805	0.8889
Gaussian NB	0.7475	0.7727	0.7222	0.75	0.7222	0.7222	0.7222
MLP Classifier	0.9773	0.9545	1	0.975	0.9474	0.973	1
LDA	1	1	1	1	1	1	1

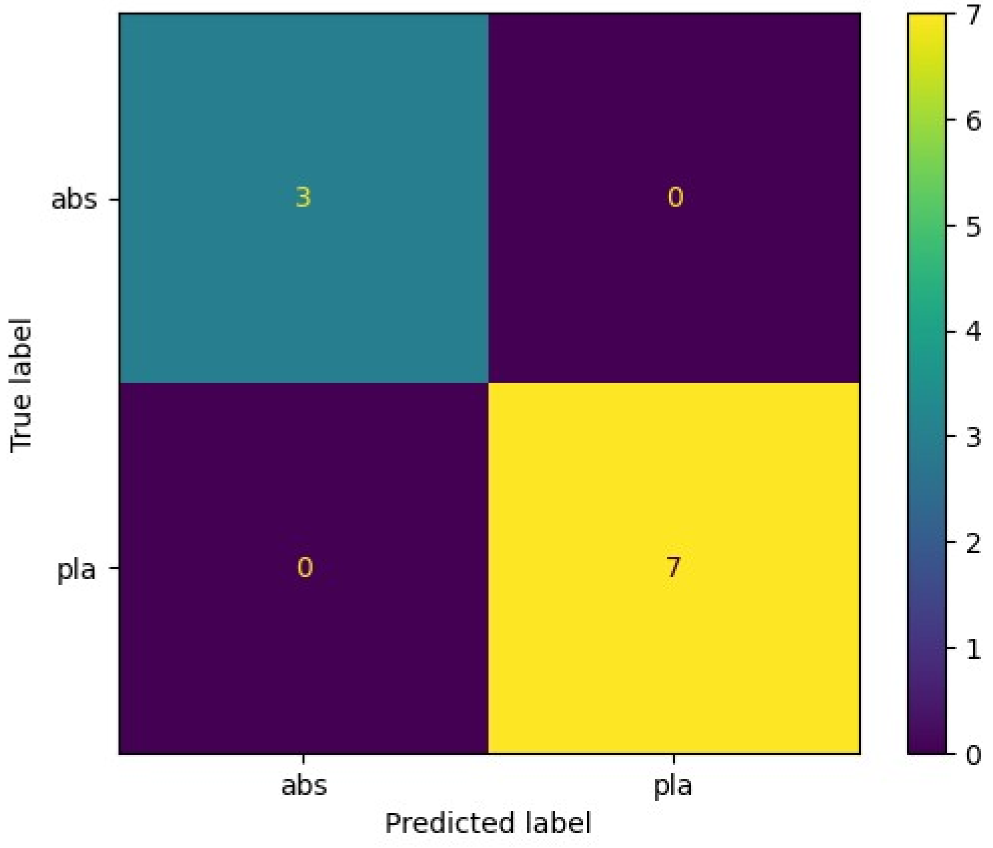

Fig-4: Confusion Matrix of Predicted Labels

References:

[1] https://www.kaggle.com/datasets/afumetto/3dprinter
[2] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html