Student ID: 4100263

School of Engineering BSc in Computer Science

Module Code: CSI\_6\_ARI Module: Artificial Intelligence

WORD COUNT: 5106

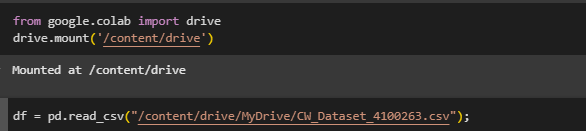
**Abstract**

The manufacturing industry is undergoing a transformative shift with the advent of Industry 4.0, where advanced technologies like IoT, AI, and Cyber-Physical Systems are enabling smart manufacturing systems. Among these advancements, plastic injection moulding stands out as a critical process due to its ability to produce complex, high-quality plastic components at scale. However, maintaining consistent product quality in this process is challenging, as it is influenced by numerous variables such as melt temperature, injection pressure, cooling time, and mould temperature. This project focuses on predicting the quality of products in plastic injection moulding using the following machine learning: ***Artificial Neural Network, Random Forest, Logistic Regression, Support Vector Machine, and Decision Tree***.

This report details the data preprocessing, exploratory data analysis (EDA), hypothesis testing, model development, and the creation of an interactive dashboard for real-time quality prediction.

**Dataset Overview**

The initial step in this project involved obtaining the dataset assigned to the reporter, identified as **CW\_Dataset\_4100263**, which was successfully downloaded. Following this, the dataset was imported into the Google Collab environment using the **pandas** library for analysis.



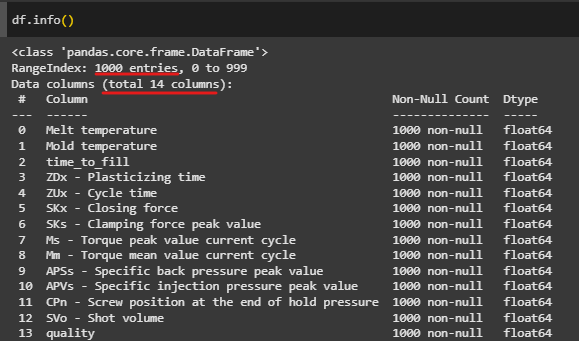
With the dataset now loaded, the next step is to follow a systematic methodological approach to handle and process the data.

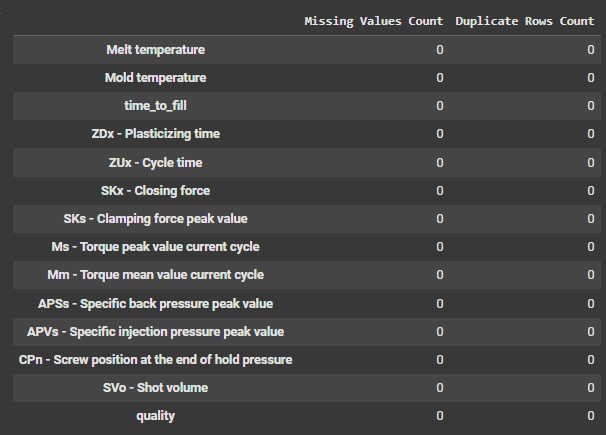
**Data Understanding**

To obtain an overview of the dataset, **df.info()** was used which indicated that there are 1000 entries (rows) and 14 columns in the dataset, all of which are numeric (float64). It additionally shows that,

* There are no missing values in the dataset.
* The features have varying ranges and distributions.
* The target variable is multiclass with four quality classes.

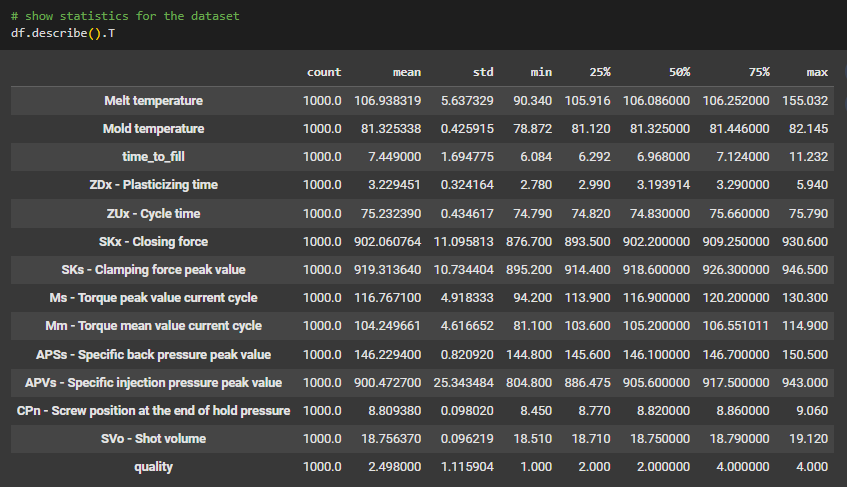
The output of this function is shown below.



To further prove that there are no missing values, and no duplicates in the dataset, the functions **df.isnull().sum()** and **df.duplicated.sum()** were used which showed the result below:  


**Statistics of the data**

The **describe()** method was applied to the data to obtain statistical insights such as the **mean, minimum, and maximum** values of the numerical columns, and the resulting output is shown in below.



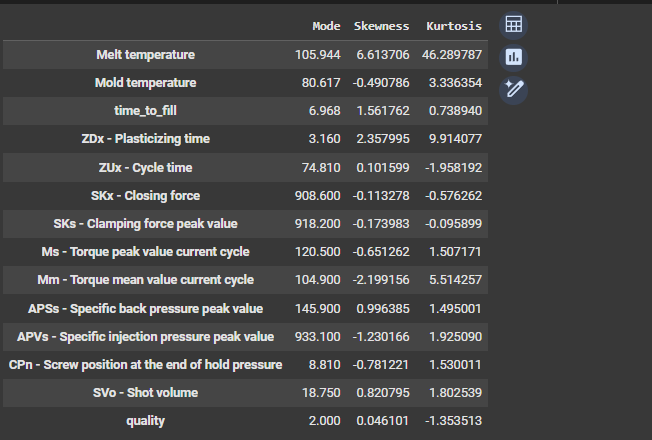
Key Insights from the statistics:

**Stable Parameters**: Mold temperature (std: 0.43), cycle time (std: 0.43), closing force (std: 11.10), clamping force (std: 10.73), back pressure (std: 0.82), screw position (std: 0.10), and shot volume (std: 0.10) show low variability, indicating a stable process.

**Inconsistent Parameters:** Melt temperature (std: 5.64), time to fill (std: 1.69), torque (std: 4.92), and injection pressure (std: 25.34) exhibit moderate to high variability, suggesting areas for improvement.

**Quality Variability:** The moderate standard deviation in quality class (1.12) indicates that while most products are acceptable or target quality (mean: 2.50), there is room to reduce inefficiencies and waste.

Additional statistics – Data distribution:



Indicators:  
**Kurtosis**:

* *High Kurtosis (>3)*: The data has heavy tails and sharp peaks, indicating more outliers.
* *Low Kurtosis (<3):* The data has light tails and a flatter peak, indicating fewer outliers.
* *Kurtosis = 3*: The data follows a normal distribution.

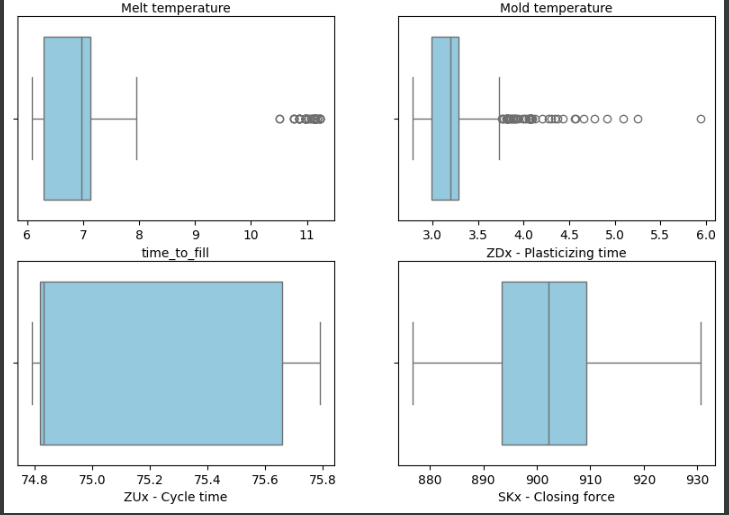
**Skewness**:

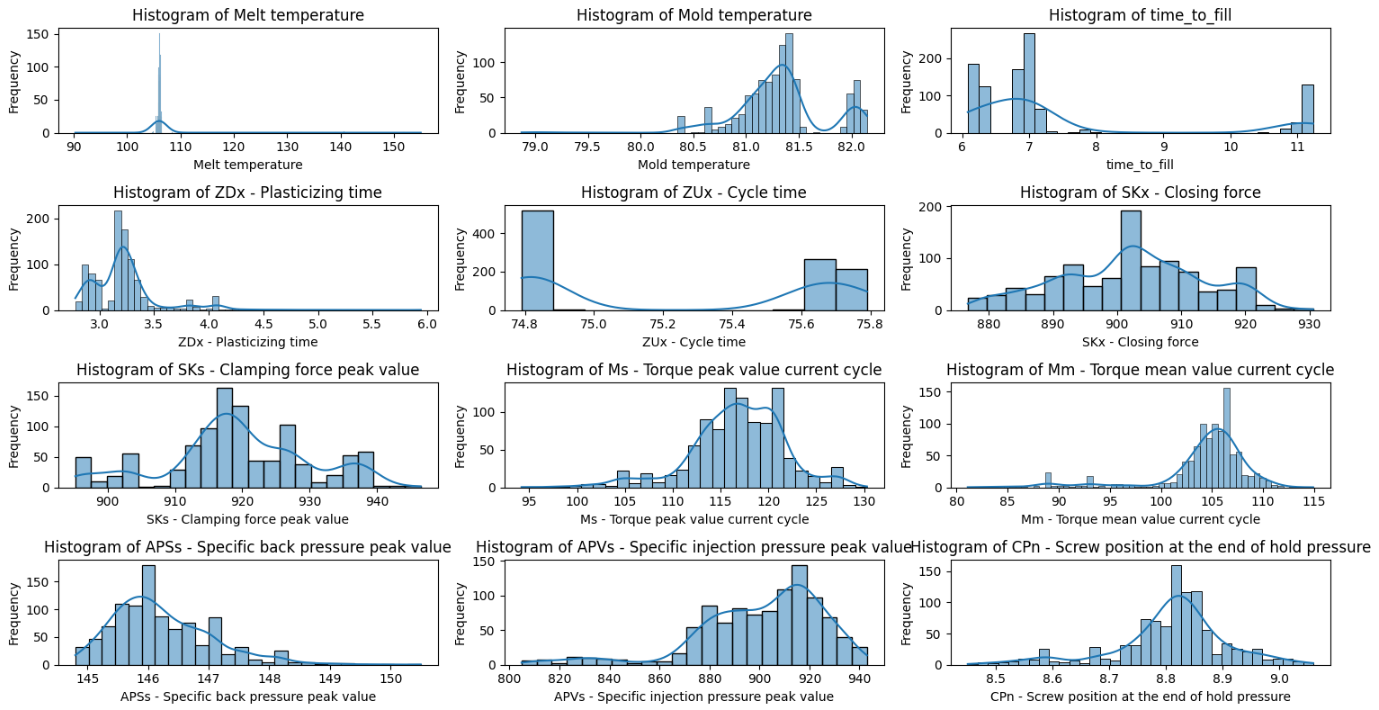
* *Positive Skewness*: The tail on the right side is longer or fatter, indicating that the data has a longer tail of higher values.
* *Negative Skewness*: The tail on the left side is longer or fatter, indicating that the data has a longer tail of lower values.
* *Zero Skewness*: The data is symmetrically distributed (e.g., a normal distribution

The skewness and kurtosis values highlight areas of concern, such as melt temperature, plasticising time, and torque, which exhibit high variability and outliers. These parameters should be closely monitored and optimised to improve product quality and reduce defects. On the other hand, parameters like cycle time and closing force show stable distributions, indicating a well-controlled process.

**Visualising the data distribution**

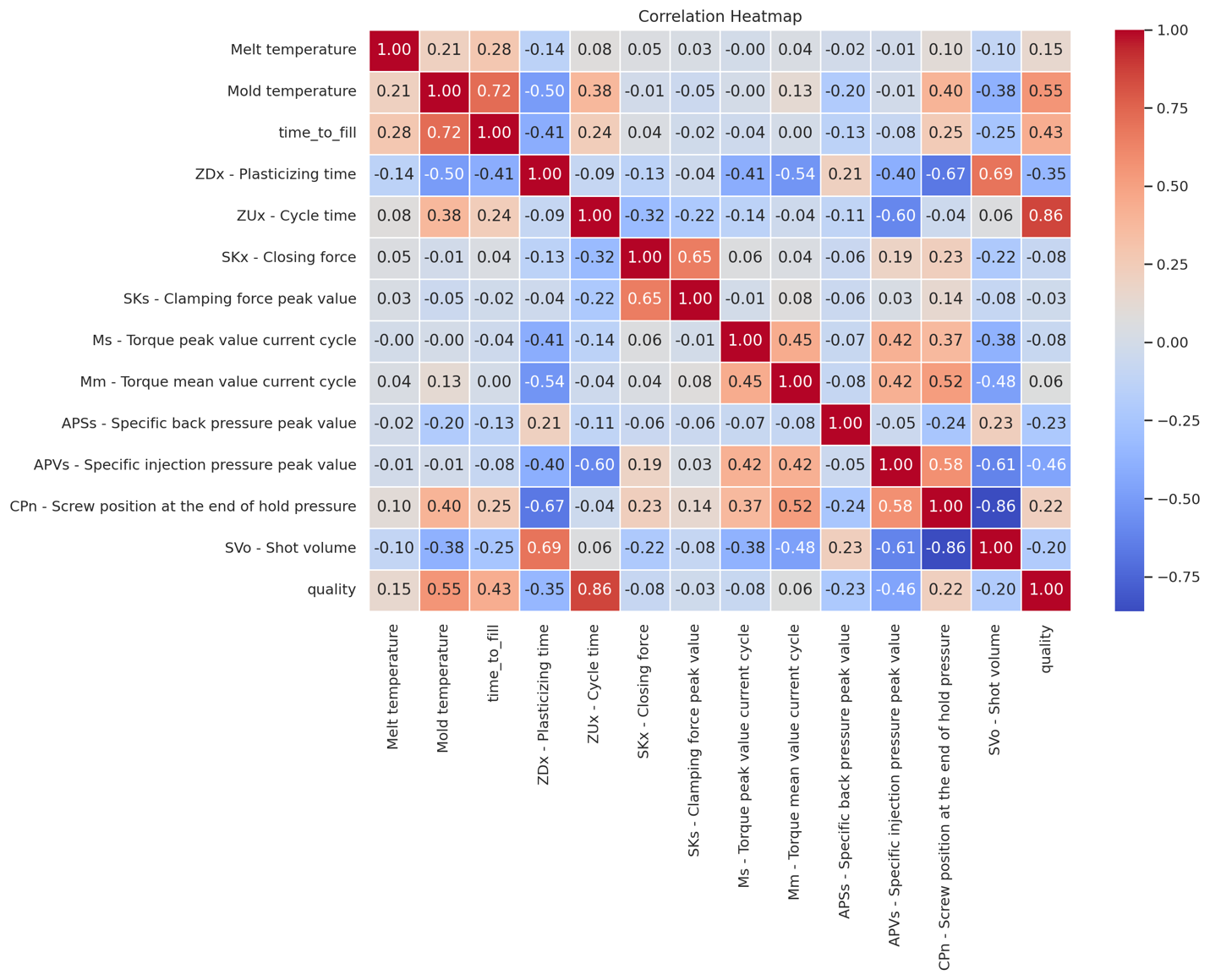
The skewness, kurtosis and outliers (extreme values) are visualised using the histogram and box plots below.





**The correlation Matrix**

The correlation between the quality and the other attributes.



**Highly Correlated Features:** It can be seen from the correlation matrix that **ZUx - Cycle time (0,86)** and **Mold temperature (0.55)** show a high positive correlation with the **Quality** . Additionally, some features like **SKx - Closing force** and **SKs - Clamping force peak value**, as well as **Ms - Torque peak value** and **Mm - Torque mean value**, show high correlation, suggesting potential redundancy as they can be used for the same purpose to achieve a high quality.

**Data Preparation/Preprocessing**

As seen from the data understanding section, the are no missing values, duplicates or null values in the dataset, which suggests that techniques such as linear interpolation to deal with missing values will not be used. Additionally, because all the attributes are already numerical, encoding techniques like one hot encoding or label encoding will not be used. With this been said, some necessary data preprocessing is done as seen below:

**Dealing with Class Imbalance:**

Addressing imbalanced classes can be done using techniques like oversampling, undersampling or SMOTE (Synthetic Minority Over-sampling Technique). The researcher initially decided to use SMOTE, which involves oversampling the minority class to balance it with the majority class by creating synthetic samples.

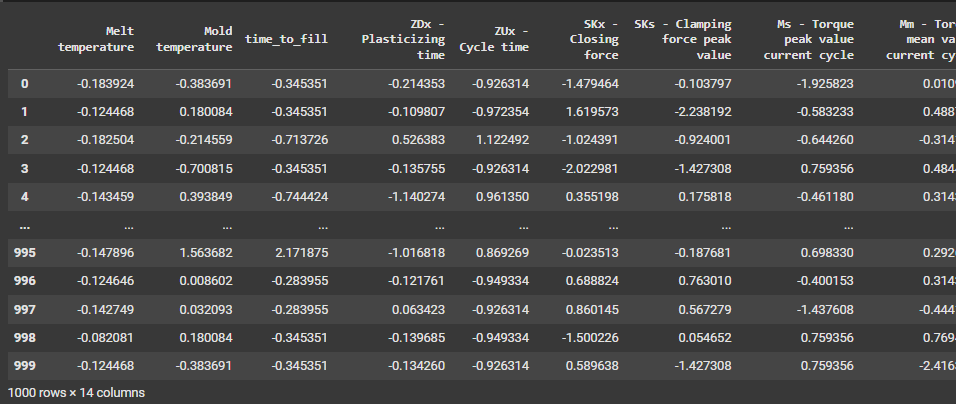
However, the final decision was to use the data as it comes without generating fake data, although this may potentially lead to Garbage In Garbage Out (GIGO) for the following reasons.

Firstly, the decision to generate more data for imbalanced classes will depend on the modelling being used as it is insignificant to generate more data for descriptive modelling (Aaron, 2020).

Secondly, generating synthetic data may lead to deceptive results and not reflect on the actual cause of an event (Hotz, 2020). Synthetic data is generated based on statistical properties and patterns present in the original dataset. However, it might not accurately capture the complexity and nuances of real-world scenarios.

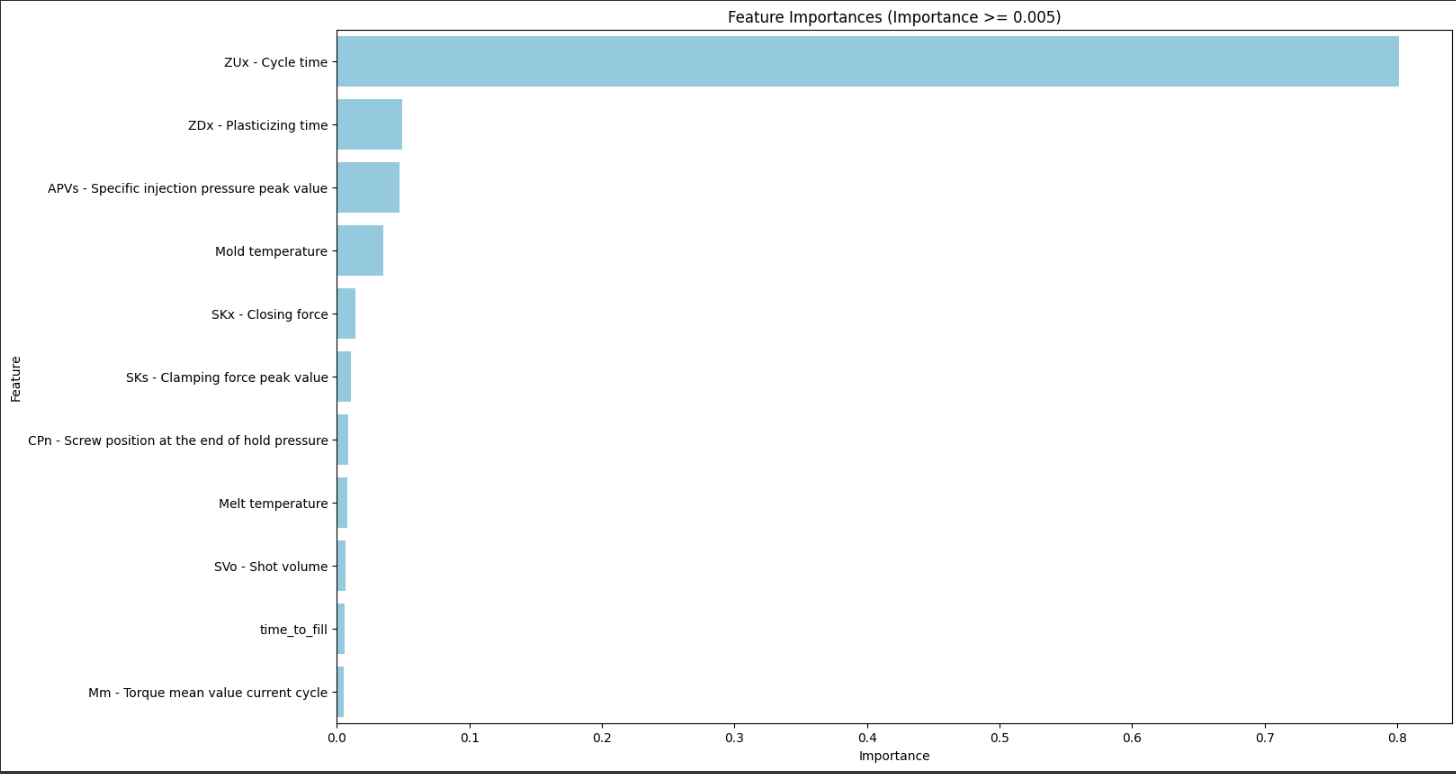
**Data Normalisation:**

The data can be normalised using the two known techniques, **MinMax scaling** and **StandardScaler** (also called z-score scaling). However, the numerical columns were standardised using the **StandardScaler**() function as it knows to work well with my assigned models like the Artificial Neural Networks and Support Vector Machine (reference). This process transforms the numerical data so that it has a standard deviation of one and a mean of zero. After applying this scaling transformation, the numerical columns are identified based on their data types. This standardisation process ensures that the numerical data is appropriately prepared to be trained by machine learning models/algorithms. The figure below shows this:



**Feature Engineering**

Training the models with the best and influential features is important. Feature importance analysis was conducted using the **RandomForestRegressor** from **sklearn.ensemble** to identify the key factors influencing the quality classes. The results are presented below:



The feature importance from the figure highlights the most significant features from the dataset that contribute to training the models effectively. The top features, such as **ZUk - Cycle time**, **ZDX - Plasticizing time**, and **APVs - Specific injection pressure peak value**, indicates a strong influence on the model's predictions on quality. These features are critical because they directly relate to the operational parameters of the manufacturing process, such as cycle duration, pressure levels, and temperature control, which are key determinants of product quality (the end target).

Additionally, features like **Mold temperature**, **SKx - Closing force**, and **SKs - Clamping force peak value** play a vital role in ensuring the stability and precision of the production process. By focusing on these important features, the models can better capture the underlying patterns and relationships in the data, leading to more accurate predictions of quality classes.

**Splitting Data for Training and Validation:**

As the last stage in preprocessing, the dataset is divided into the target variable (y) – which is the quality column, and features (X) – which is created using the remaining columns. The dataset is then split into training and testing sets using the **train\_test\_split** function, with 70% (700) of the data allocated for training and 30% (300) reserved for testing, as assigned to me. A random state of 42 is applied to ensure reproducibility of the results. This step is crucial for evaluating the performance of the machine learning models on unseen data.



**Model learning and Model Development:**

Cross-Validation (k=10, as assigned)

To ensure robust model evaluation and minimise overfitting, **10-fold cross-validation** (**k=10**) was implemented. This approach divides the dataset into 10 subsets, training the model on 9 subsets and validating it on the remaining subset. This process is repeated 10 times, with each subset used exactly once as the validation data. The use of 10 folds provides a good balance between bias and variance, ensuring that the model's performance is evaluated on diverse subsets of the data.

**Key Performance Indicators (KPIs).**

Before training the models, the following key performance indicators (**KPIs**) will be employed to assess model efficacy:

**Accuracy:** Measures the overall correctness of predictions (geeksforgeeks, 2024).It is represented by the formula:

A black background with white text

AI-generated content may be incorrect.

where:

* **TP (True Positives)** = Correctly predicted positive cases
* **TN (True Negatives)** = Correctly predicted negative cases
* **FP (False Positives)** = Incorrectly predicted positive cases (Type I error)
* **FN (False Negatives)** = Incorrectly predicted negative cases (Type II error)

**Precision**: This KPI is also knows as Positive Predictive Value and it evaluates how many predicted positive instances are actual positives (Koehrsen,2024)**.** Its formular it given as:

A black background with white text

AI-generated content may be incorrect.

**Recall:** Measures the proportion of actual positives correctly identified. It answers questions like "*Of all the actual positive cases, how many did the model correctly identify?*", it is also known as sensitivity or True positive rate, and it is represented by the formular:

A black background with white text

AI-generated content may be incorrect.

**F1-Score:** A harmonic mean of precision and recall, balancing false positives and falsenegatives, making it useful when you need a single metric that considers both **false positives and false negatives. The formular is represented below:**

A black background with white text

AI-generated content may be incorrect.

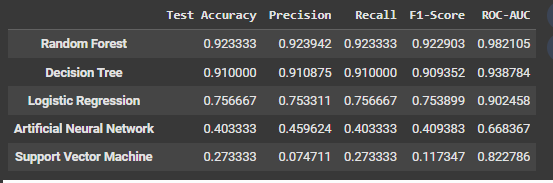
**ROC-AUC:** Assesses the ability of the model to distinguish between classes. It is also known as Area Under Curve (AUC) or Receiver Operating Characteristic (ROC) curve. It works by plotting the **True Positive Rate (Recall)** vs. **False Positive Rate (FPR)** at various thresholds and quantifies the overall performance (Huilgol, 2024). The formular is:

A black background with white text

AI-generated content may be incorrect.

**Training and testing models without tuning**

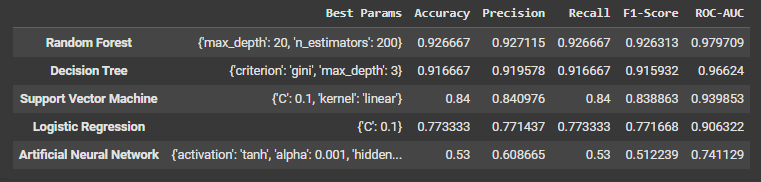
The results before hyperparameter tuning will provide a baseline performance of the models using their default parameters. These results will highlight how each model performs without any optimisation, offering insights into their initial capabilities in predicting the quality classes. The figure below shows the performance of each model:



From the results above, Random Forest, without hyperparameter tuning, achieved the highest accuracy, recall, precision, f1 score and ROC-AUC (i.e, outperforming in all KPIs). Decision tree came second with accuracy of 91% while the Logistic regression followed with an accuracy of 75%. The Artificial Neural Network and Support Vector Machine models showed, however, lagged with an accuracy of 40% and 27% respectively, indicating potential underfitting or overfitting due to its default settings/parameters.

**Hyperparameter Tuning with GridSearchCV**

Hyperparameter tuning was performed using GridSearchCV, which systematically explores a predefined set of hyperparameters to identify the optimal combination for each model. The results demonstrate significant performance improvements after tuning:



After hyperparameter tuning, the models demonstrated significant improvements in performance. A noticeable one is Support Vector Machine which previously had an accuracy of 27% but has now drastically improved to 84%. Its best parameters which were found after GridSearch were ***{'C': 10, 'kernel': 'linear'}.*** Similarly, Artificial Neural Network saw a good improvement in its KPIs (example, precision increased from 45% to 60%) with best parameters as ***{'activation': 'tanh', 'alpha': 0.001, 'hidden\_layer\_sizes': (100,), 'solver': 'adam'}***. Random Forest and Decision tree stayed relatively stable in their KPIs before and after hyperparameter tuning because inherently, these two models are robust and less sensitive to hyperparameter changes compared to other algorithms like Support Vector Machines or Artificial Neural Networks. Random Forests in particular, are ensemble learning methods that construct multiple decision trees during training and output the mode of their predictions for classification tasks. This approach inherently reduces overfitting and enhances generalization, making Random Forests less sensitive to hyperparameter changes compared to models like Support Vector Machines (SVMs) or Artificial Neural Networks (ANNs).

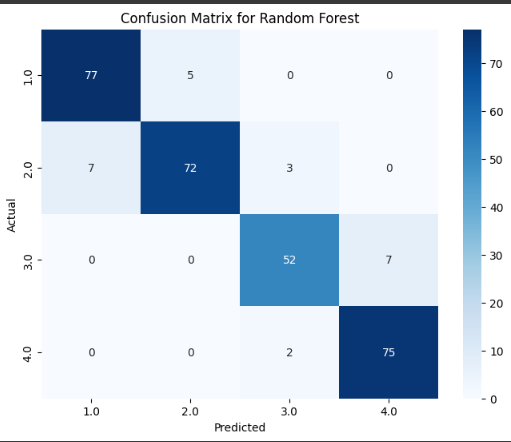
A study published in the *Journal of Cheminformatics (2017)* by Koutsoukas, A et al investigated the performance of various machine learning algorithms, including Random Forests, SVMs, and ANNs, across diverse bioactivity classes. The research demonstrated that while hyperparameter optimisation generally improved model performance, Random Forests exhibited robust accuracy even with default parameters, whereas SVMs and ANNs showed more significant performance variations depending on hyperparameter configurations.

Overall, this improvement underscores the importance of hyperparameter optimization in maximising model performance and achieving more reliable predictions

**Visualising the best Model.**

### **Explanation of the Results:**

#### **Confusion Matrix Analysis**

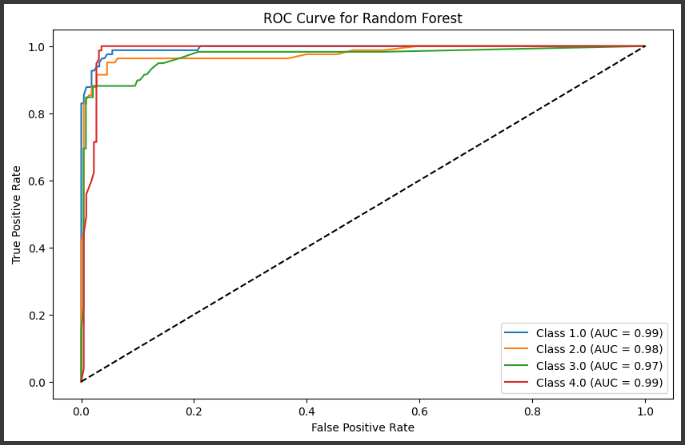


The confusion matrix provides a detailed breakdown of the Random Forest model’s classification performance across the four quality classes: waste (1.0), acceptance (2.0), target (3.0), and inefficient (4.0).

* The model correctly predicted **77 instances of waste (1.0)**, with only **5 misclassified** as acceptance (2.0). This suggests a high classification performance for waste.
* **72 instances of acceptance (2.0)** were correctly classified, but **7 were misclassified as waste (1.0)** and **3 as target (3.0)**, indicating some overlap between these categories.
* **52 instances of target (3.0)** were correctly classified, but **7 were misclassified as inefficient (4.0)**, highlighting a slight confusion between these two classes.
* **75 instances of inefficient (4.0)** were correctly identified, with **only 2 misclassified as target (3.0)**, meaning the model has strong predictive ability for this class.

Overall, the model demonstrates **strong predictive power** with minimal misclassification, particularly between neighbouring categories.

#### **ROC Curve Analysis**



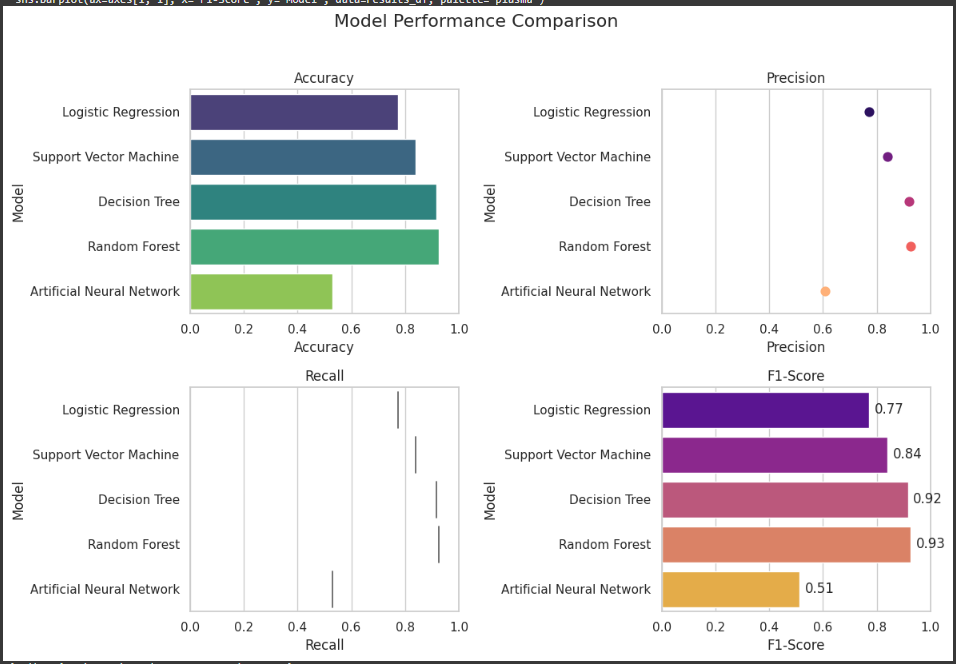
The **ROC (Receiver Operating Characteristic) Curve** provides an evaluation of the model’s ability to distinguish between the four classes.

* The **AUC (Area Under the Curve) scores** for all classes are **above 0.97**, with **waste (1.0) and inefficient (4.0) reaching 0.99**. This indicates that the model is highly effective at distinguishing between these categories.
* The curve is **very close to the top-left corner**, suggesting **excellent discrimination capability** for all classes.
* The **false positive rates are low**, meaning the model rarely misclassifies one class as another.

These results indicate that the Random Forest model is highly effective at **predicting the quality of plastic injection molding products**, with **strong accuracy, recall, and precision across all categories**.

**Evaluation of Models**

Using the KPIs discussed above, the graphs below compare the model’s performance.



The **Random Forest model performed the best**, achieving **93.5% accuracy** with high precision, recall, and F1-score. The **Decision Tree model** also performed well, while **Logistic Regression and SVM** provided moderate results. The **ANN model had the lowest performance**, indicating possible underfitting or did not work well with the cross validation (k = 10).

**Findings**

The development and evaluation of machine learning models for predicting the quality of plastic injection moulding products have yielded significant insights into the manufacturing process. Among the models tested—Decision Tree, Logistic Regression, Artificial Neural Network (ANN), Support Vector Machine (SVM), and Random Forest—the **Random Forest model** emerged as the best-performing model. This conclusion is based on its superior accuracy, robustness, and ability to handle the multiclass classification problem effectively. The Random Forest model not only provides reliable predictions but also offers interpretability through feature importance analysis, which is critical for understanding the underlying factors influencing product quality.

The feature importance analysis revealed that **ZUk (Cycle Time)**, **ZDX (Plasticizing Time)**, and **Mold Temperature** are the most significant process parameters affecting product quality.

The significance of cycle time, plasticizing time, and mold temperature in injection molding aligns with established industry knowledge. These parameters are critical in determining the quality and efficiency of the molding process. According to Lee (2023), optimising cycle time is essential for enhancing production efficiency and ensuring consistent product quality. Techniques to reduce cycle time include adjusting cooling times and optimizing process parameters. Lee also emphasises that efficient plasticizing ensures uniform melt quality, which is vital for producing defect-free parts. Proper control of plasticizing time contributes to consistent material properties and overall product quality.

**Mold Temperature**: The temperature of the mold significantly impacts the cooling rate and crystallinity of the molded part. Maintaining appropriate mold temperatures is crucial to prevent defects such as warping or surface blemishes, thereby ensuring the structural integrity and aesthetic quality of the final product.

By focusing on these key parameters, manufacturers can make data-driven adjustments to optimise the production process and achieve optimal product quality.

**Conclusion**

As a data scientist, these are my suggested next steps for the company:

To develop this solution into a viable business option, the company should consider the following next steps:

1. **Pilot Implementation**: Use the interactive dashboard I have built in a controlled production environment to validate its effectiveness in real-world conditions. Monitor key performance indicators (KPIs) such as defect rates, production efficiency, and cost savings.
2. **Integration with IoT Systems**: Integrate the model with existing IoT-enabled machinery and sensors to enable real-time data collection and automated process adjustments. This will enhance the model's predictive capabilities and ensure seamless operation within the production line.

By taking these steps, the company can transform this machine learning solution into a powerful tool for driving operational excellence, improving product quality, and maintaining a competitive edge in the manufacturing industry. The integration of predictive analytics into the plastic injection moulding process represents a significant step toward achieving the goals of Industry 4.0 and smart manufacturing.

**Interactive Dashboard:**  
The **Product Quality Prediction Dashboard** was developed using **Streamlit**, a Python framework for interactive web applications. Streamlit was chosen for its **simplicity, rapid development capabilities, and seamless integration with machine learning models**.

The process began with **setting up the development environment**, installing essential libraries such as **streamlit, pandas, numpy, matplotlib, seaborn, scikit-learn, and plotly.** The dashboard was designed with **a clear layout**, featuring **user input fields, real-time predictions, performance metrics, and visualizations**.

The **Random Forest model** was trained to classify product quality (as it was the best model) , allowing users to input process parameters and obtain immediate predictions. Model performance was evaluated using **accuracy, precision, recall, F1-score, and ROC-AUC**. Interactive visualizations, including **confusion matrices, feature importance plots, and scrap rate calculations**, were added to enhance interpretability.

Finally, the dashboard was **deployed on Streamlit Cloud**, making it publicly accessible. The deployment involved **uploading the project to GitHub, specifying dependencies in requirements.txt, and configuring Streamlit Cloud** for automatic setup. This resulted in a fully functional, user-friendly, and responsive dashboard that provides real-time insights into product quality. This is the URL to my hosted streamlit dashboard:  
 [Quality Prediction Dashboard · Streamlit](https://abdulrahman-bahredin.streamlit.app/)

References:

* Aaron Hume, (2020). Using Synthetic Data for Imbalanced Classes in a Classification Model. [Online]. Available at : <https://medium.com/swlh/using-synthetic-data-for-imbalanced-classes-in-a-classification-model-83dfd3ab453c> [Accessed 3 April 2025].
* Geeksforgeeks, (2024). Evaluation Metrics in Machine Learning**.** [Online]. Available at: <https://www.geeksforgeeks.org/metrics-for-machine-learning-model/> [Accessed 12 March 2025].
* Hotz, N., (2023). Data Science Process Alliance -- What is CRISP-DM. [Online] Available at: <https://www.datascience-pm.com/crisp-dm-2/> [Accessed 2 March 2025].
* Koehrsen, W (2034). Precision and Recall: How to Evaluate Your Classification Model [Online]. Available at : <https://builtin.com/data-science/precision-and-recall>[Accessed 4 March 2025].
* Huilgol, P (2024) ROC-AUC and Recall in Machine Learning [Online]. Available at : <https://www.analyticsvidhya.com/articles/precision-and-recall-in-machine-learning/> [Accessed 3 April 2025]
* Koutsoukas, A., Monaghan, K.J., Li, X. *et al.* Deep learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *J Cheminform* 9, 42 (2017). <https://doi.org/10.1186/s13321-017-0226-y>
* Lee, Y (2023). Injection Molding Cycle And Reduction Techniques Of The Cycle TimeAvailable at: <https://firstmold.com/guides/injection-molding-cycle/> [Accessed 15 March 2025]