SmartBuild

Alek Cvetanoski, Mahnoor Azam, Tringë Gruda
IBS5
21.12.2023

Question Definitions

- Q1: Create a predictive model for the attribute "weight_in_kg"
- Q2: Create a predictive model for the "error" attribute
- Create predictive models for further product characteristics (different models, type of model, or feature sets)

Take Home Message

- It is possible to predict error with the highest accuracy of 92%
- It is possible to predict whether there is an error or not before production
- Predicting weight in kg and error with high accuracy can optimise manufacturing processes, control costs and ensure quality. Preventing errors before production can save the company up to 86.67% in costs

Approach to Findings

- Download and review data
- Check data quality
- Select the appropriate models to be used
- Program the first models
- Review and edit code
- Review results

Reason for the analysis

- Production Optimisation
- Identify potential errors - manage and mitigate risks
- Operational excellence within SmartBuild
- Waste reduction

Overview

- Q1: weight_in_kg ⟶ RandomForestRegressor
- Q2.1: error⟶ XGBClassifier, RandomForestClassifier
- Q2.2: error_type ⟶ RandomForestClassifier
- Q3: Quality ⟶ RandomForestRegressor
- Q4: reflectionScore ⟶ LinearRegression, ConfusionMatrix
- Q5: nicesness ⟶ LinearRegression, DecisionTreeRegression, XGBRegression

Q1: weight_in_kg

Model details:

RandomForestRegressor

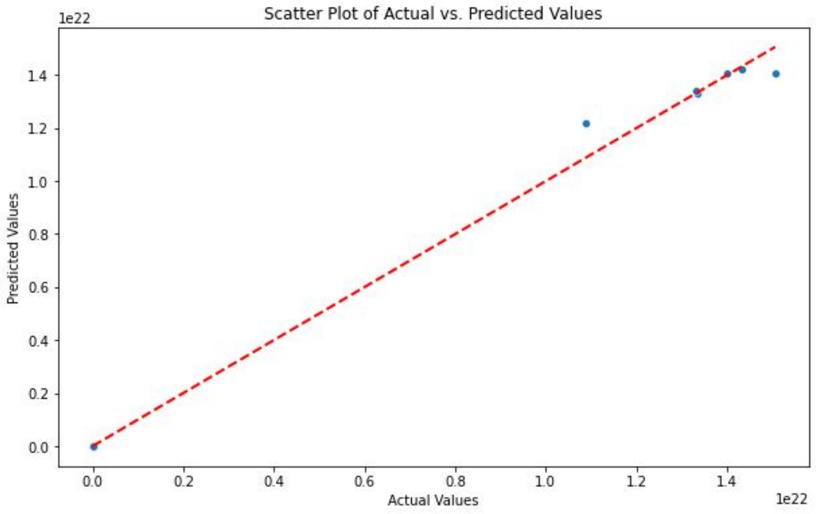
Data inputs: width, height, pressure, karma, modulation

Observations:

Random Forest Mean Squared Error: 1.24e+39

Random Forest R-squared: 0.997

Weight in kilograms can be predicted with high accuracy when we consider the following features:



Feature	Width	Height	Pressure	Karma	Modulation
Importance	0.996242	0.003379	0.000070	0.000163	0.000145

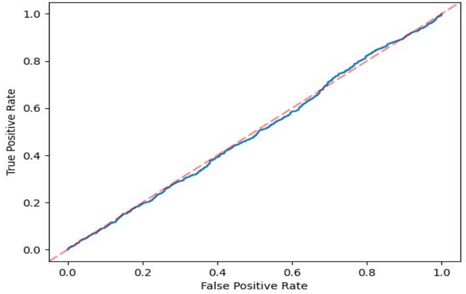
Q2.1: error

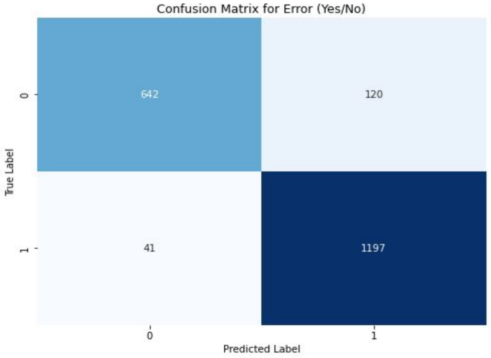
Model details:

XGBClassifier

RandomForestClassifier

Data inputs: width, height, pressure, karma, modulation, ionizationclass_A, ionizationclass_B, ionizationclass_C

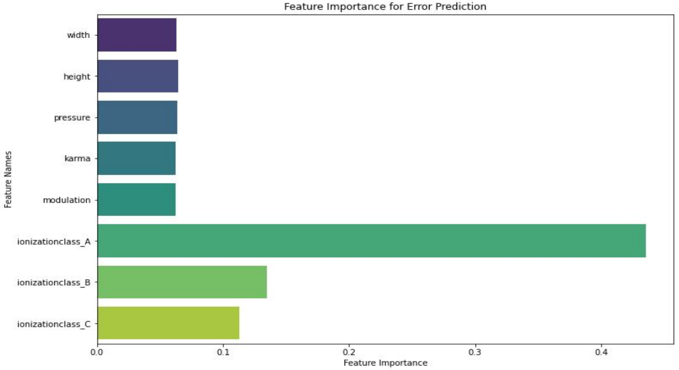




Observations:

Accuracy: 0.92

The models can accurately predict 92% of the time whether there is an error or not.



Q2.2: error_type

Model details:

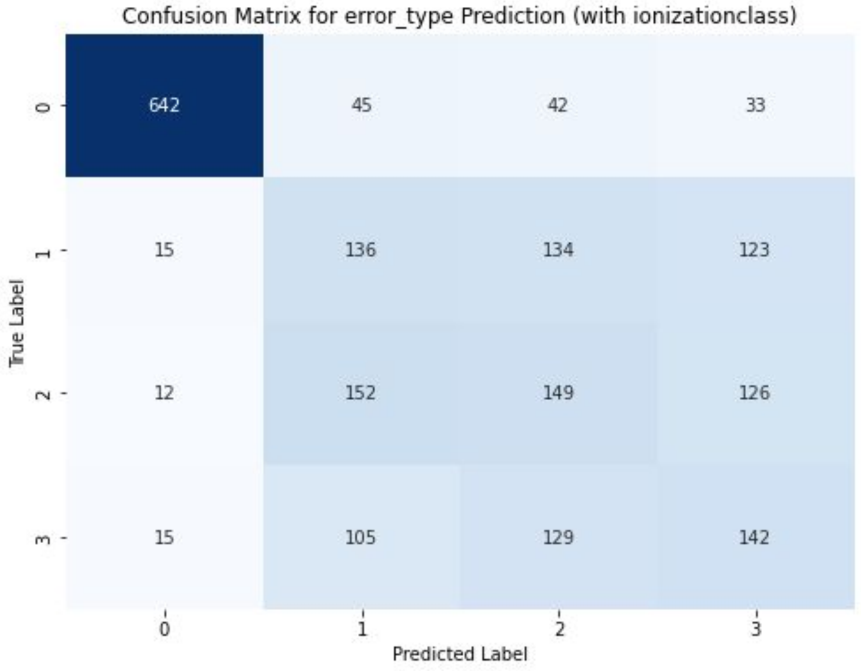
RandomForestClassifier

Data inputs: width, height, pressure, karma, modulation, ionizationclass_A, ionizationclass_B, ionizationclass_C

Observations:

Accuracy: 0.53

The model cannot predict the type of the error, but rather when there is no error at all.



Q3: Quality

Model details:

RandomForestRegressor

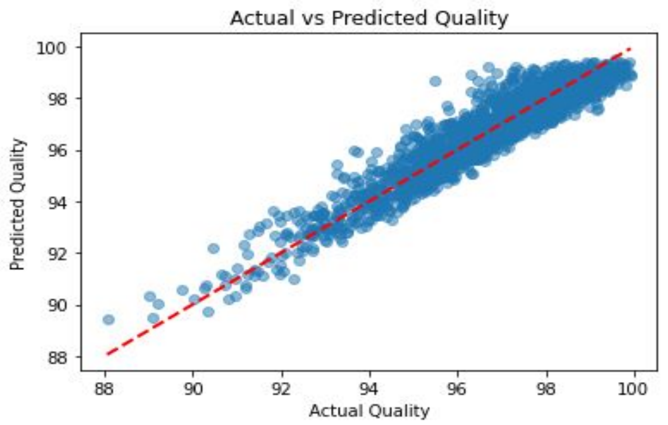
Data inputs: width, height, pressure, karma, modulation

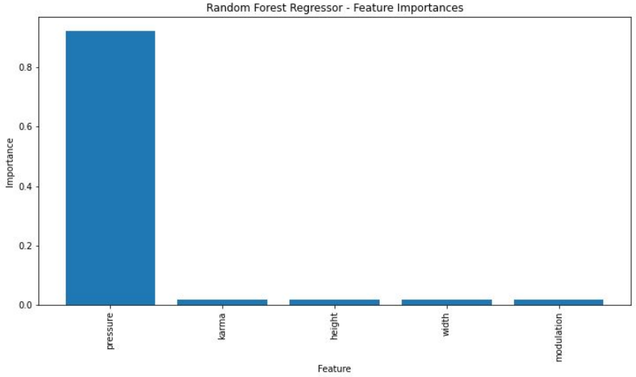
Observations:

Mean Squared Error (quality): 0.39193227994363017

R-squared (quality): 0.8912564103523382

The model can accurately predict the quality of the product based on width, height, pressure, karma, modulation, with pressure having the highest impact.





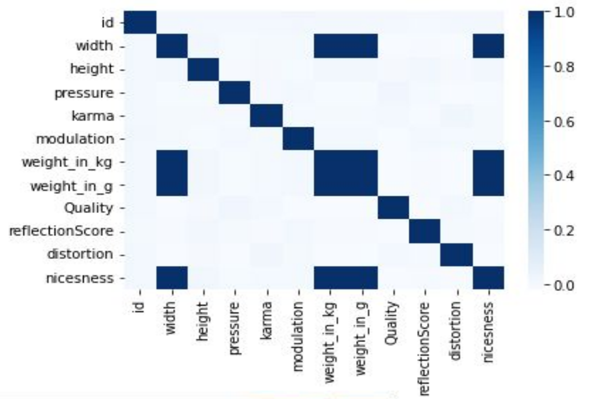
Q4: reflectionScore

<u>Model details:</u>

LinearRegression

Correlation Matrix

Data inputs: width, height, pressure, karma, modulation

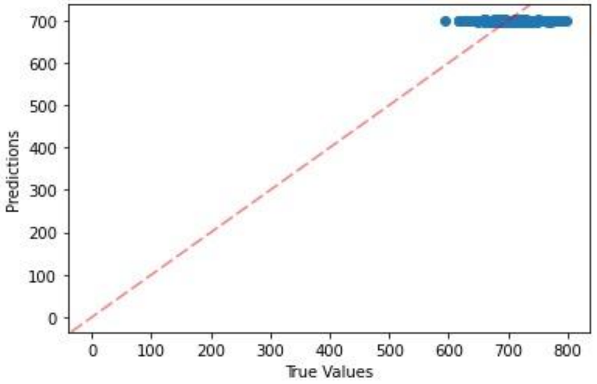


<u>Observations:</u>

Mean Squared Error: 883.3759578365087

Mean Absolute Error: 23.79509438089563

The model cannot predict the reflectionScore because the reflectionScore is not correlated to the other attributes.



Q5: nicesness

Model details:

LinearRegression

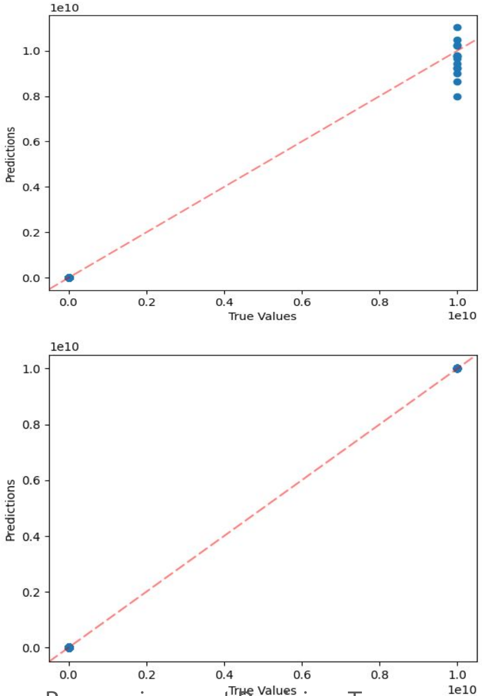
DecisionTreeRegression

XGBRegressor

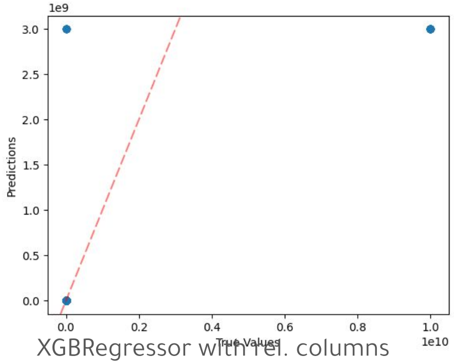
Data inputs: width, height, pressure, karma, modulation

Observations:

- The model cannot predict the nicesness because the value is subjective and there is a lack of correlation with the other attributes.
- High variance between values



Linear Regression and Decision Tree



XGBRegressor with rel. columns

Reflection - Implications for SmartBuild

- More data could be collected before the beginning of the production, enabling a larger amount of predictions

  → ensure higher accuracy for predictions
  → optimise production in the future

- For the predictions to be put in use, caution must be practised as priority

Questions?