Steel Plate Fault Detection

End-to-end Cloud Solution

Applied Machine Learning and Data Engineering in Business Context

Alisa Ilina	141804
Anurag Singh	142859
Eirik Egge	141164
Henry Stoll	141636
Magnus Eliassen	141855
Olivia Lundholm	141607

XYZ Products

Executive Summary

Delivering value across the production line with cloud based fault detection

1

Identifying the Problem

The most common production faults Bumps, K-Scratches and Z-Scratches are mainly impacted by conveyor length, steel plate thickness, and the steel type

2

Automating Fault Detection

XGBoost is correctly classifying over 80% of all faults and can be used in production to to turn manual fault classification into automated fault detection

3

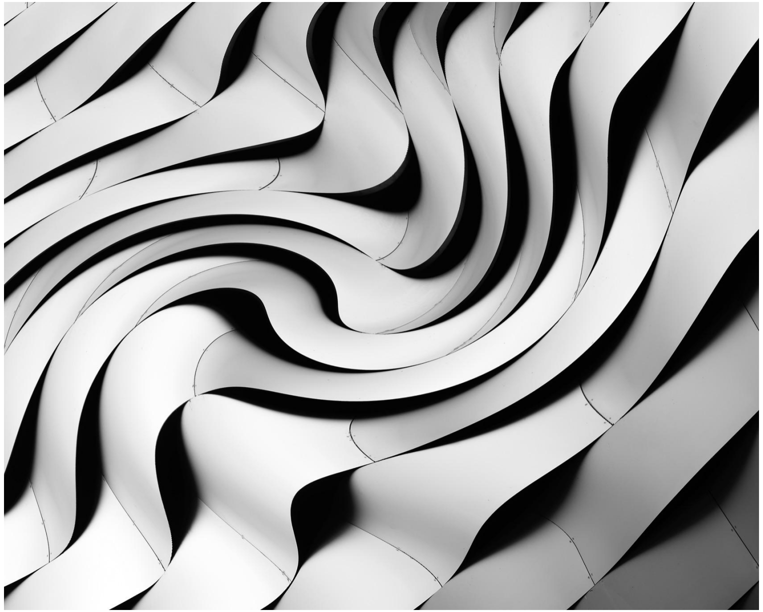
Moving to the Cloud

Cloud architecture leverages existing data sources to enable fault detection and continuous flow of data-driven insights into the production process

4

Scaling for the Future

Moving beyond analysis to scaling automated fault diagnosis across the production line whilst enabling future high value use cases



XYZ Products

Current Situation

Implementing automated fault detection at scale in XYZ Products

XYZ Products is a steel manufacturer based in Northern Sweden with a budgetary goal of reducing costs related to production faults

- ➢ Ineffective **manual quality control** system in place with high associated variable costs
- ➢ Faulty steel plates detected and returned by customers resulting in **low customer satisfaction**, loss of market share, and worsened reputation
- ➢ Costly manual collection of fault data and labeling of fault types
- ➢ Decision making related to fault prevention is done based on **human assumptions** rather than data-driven insights
- ➢ Below average OEE[1] of 60%

Key Challenges

- ➢ Manual fault detection hinders production and is **costly**
- ➢ **Lack of production insights** on the origin of faults results in poor strategic decisions
- ➢ On-premise **architecture does not accommodate for an automated solution** and fault pattern analytics
- ➢ Interrupted dataflow limits data-driven actions

Proposed Solution

End-to-end cloud solution for fault detection and prevention enables *XYZ Products* to reduce the financial impact of production faults and make data-driven decisions

Increased Quality → Increased OEE[1] → Reduced Variable Cost

[1]OEE = Operational Equipment Efficiency

XYZ Products

Approach

Ensuring optimal delivery throughout the project lifecycle

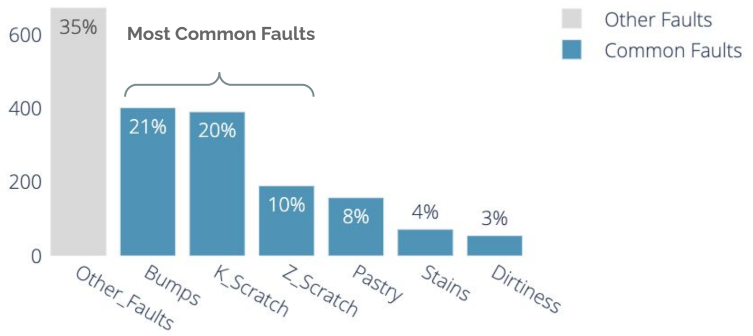
	Assessment	Development	Monitor & Scale
Time Estimate	5 Weeks	8 Weeks	18 Months
Activities	Assess the current manual process and perform fault pattern analytics Establish to-be state Perform Gap analysis Evaluate fault detection solution based on Complexity/Value trade-off	Set up IoT edge connector Set up and schedule ETL pipelines and develop ML models Build test environment Set up CI/CD and MLOps practices Execute in staging and production environment	Release strategy & monitor the maturity of developed solution Change management & training Continuous observability Run CI/CD and DevOps practices for agility
Deliverables	GAP ANALYSIS USE CASES ROADMAP	BUILD USE CASE ETL PIPELINES	AUTOMATED TESTING MONITORING SET-UP DOCUMENTATION

Data Insights

Steel type, thickness and conveyor length have the highest impact on fault type...

What are the most common faults?

Frequency of Fault Types

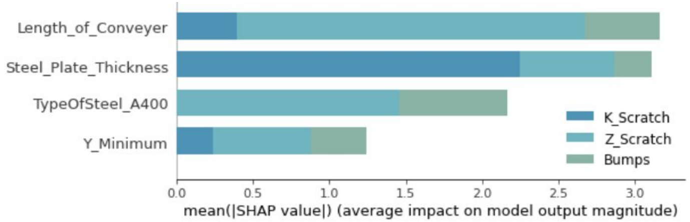


Other Faults is the most frequent class, as it accounts for 35% of the faults

Bumps, K-Scratches and Z-Scratches are the most common faults accounting for 51% of the faults

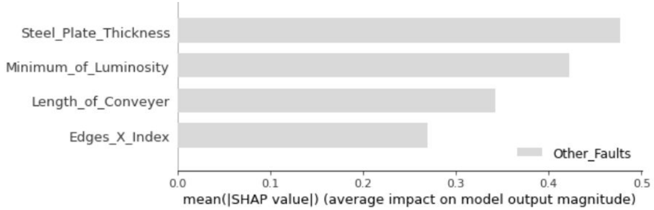
What is causing Most Common Faults?

The most common faults are mainly impacted by the features **Length of the conveyer, Steel Plate Thickness,** and the **Type of Steel**



What is causing Uncommon Faults?

Other Faults are also mainly impacted by the features **Length of the conveyer** and **Steel Plate Thickness**

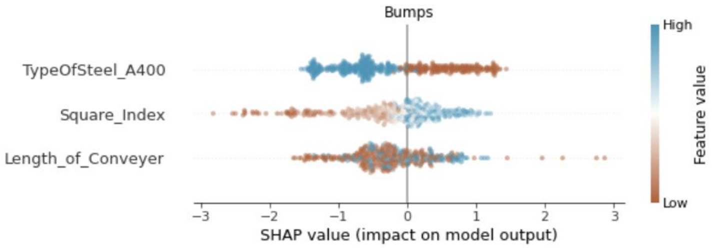


XYZ Products

… and cause most of common and uncommon faults

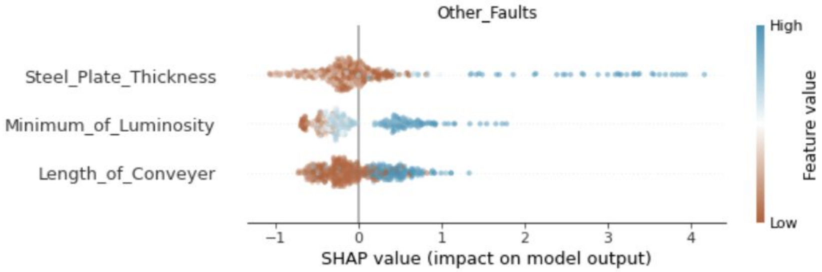
Bumps

Primarily caused by a **high Square index** and **the A300 steel type**. This steel type in combination with a **high length of the conveyer** is causing the largest portion of the faults



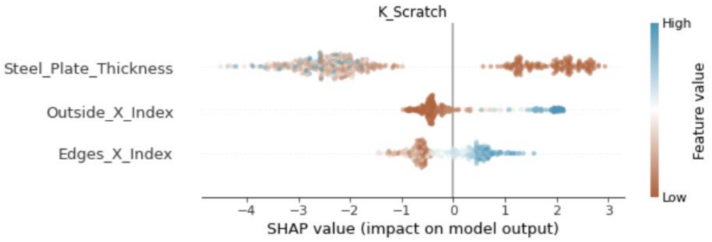
Other Faults

Primarily caused by a **high Steel Plate Thickness, high luminosity** and **high length of the conveyer**



K-Scratches

Mainly caused by **low Steel Plate Thickness**, **High Outside X Index**, and **high Edges-X-Index**

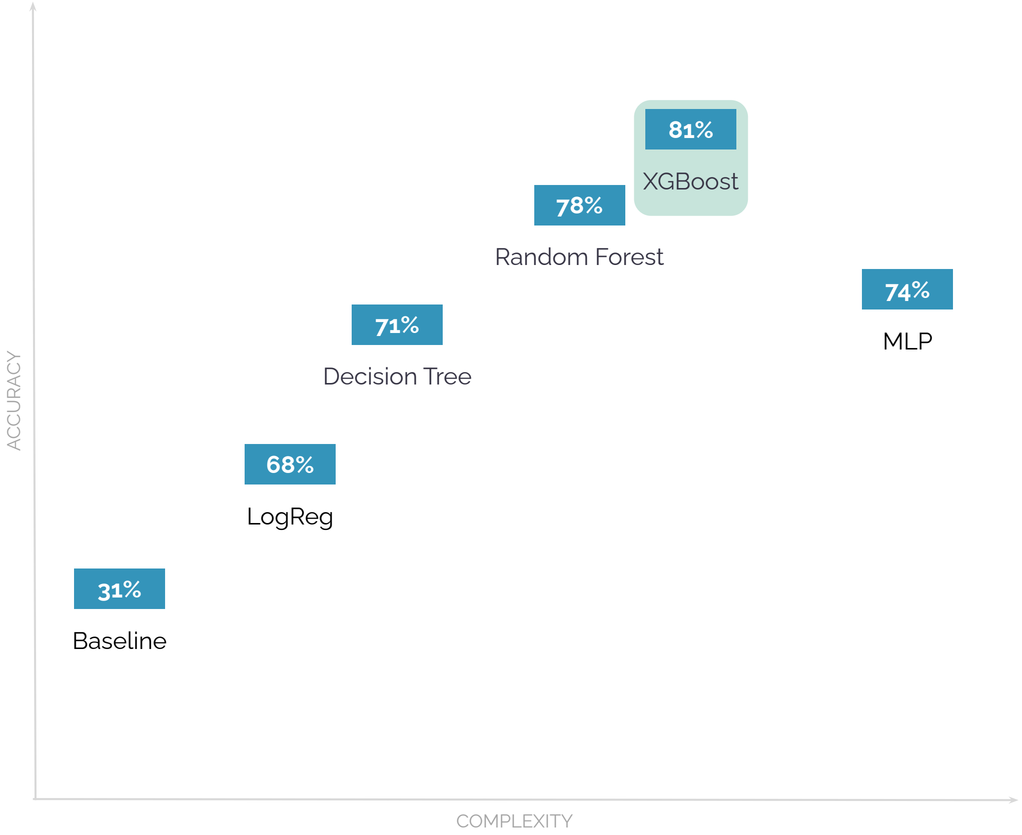


Key Takeaways

Using **A400 steel** would reduce Bumps which is the most common fault type

Higher Steel Plate Thickness would reduce the total number for K-scratches and Other Faults

XGBoost achieves highest accuracy for all fault types



ML algorithms tuned with **hyperparameters** were run to identify best performing models, **cross validation** was used in each

- ➢ **Oversampling** was implemented to avoid incorporating bias towards majority class, resulting in higher accuracy
- ➢ Experiments with and without **outliers** were run, keeping non-extreme outliers rendered better results
- ➢ **Feature selection** was implemented, strongly correlated features were dropped reducing model complexity
- ➢ **Robust scaling** was used to facilitate feature interpretation by distance-sensitive models

Multi-Class Classification

7 label classification performs only slightly worse than binary suggesting stronger class separation patterns and potential subclusters to be identified in “Other” faults

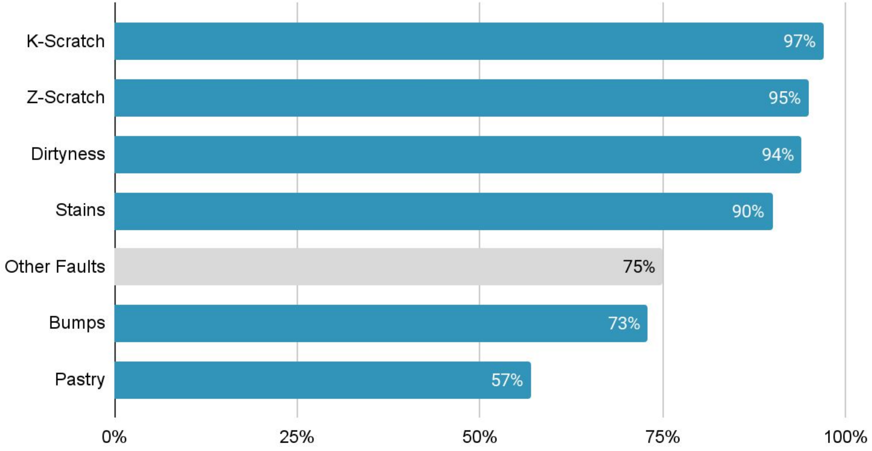
XGBoost is ready for integration into production environment

XGBoost is a widely adopted and scalable machine learning algorithm combining decision trees

- ➢ Low time and memory complexities
- ➢ Outstanding performance handling large, noisy and imbalanced datasets
- ➢ Explainable due to built in feature importance tools

Decision trees are built sequentially by minimizing errors from previous models and boosting influence of best models

XGBoost - Prediction Accuracy



XGBoost is production ready:

- ➢ The model captures over 95% of all scratches
- ➢ The model can handle both common and uncommon faults with acceptable to high accuracy

By integrating this model into production **manual fault classification** can be transformed into **automated fault detection** enabled **at scale** by **cloud architecture**

Process

Reduce costs by automating fault detection with Machine Learning...

Customer Order	Production Line	Fault Detection	Shipment
Incoming steel plate orders from happier customers	Faulty steel plates returned into relative production line stages once detected	Replace manual labelling with automatic fault detection Use machine and production data Dynamic flow of insights back into production line allows for preventive maintenance and data-driven decisions	Shipping quality-assured steel plates results in happier customers, improving reputation and reducing costs

insights

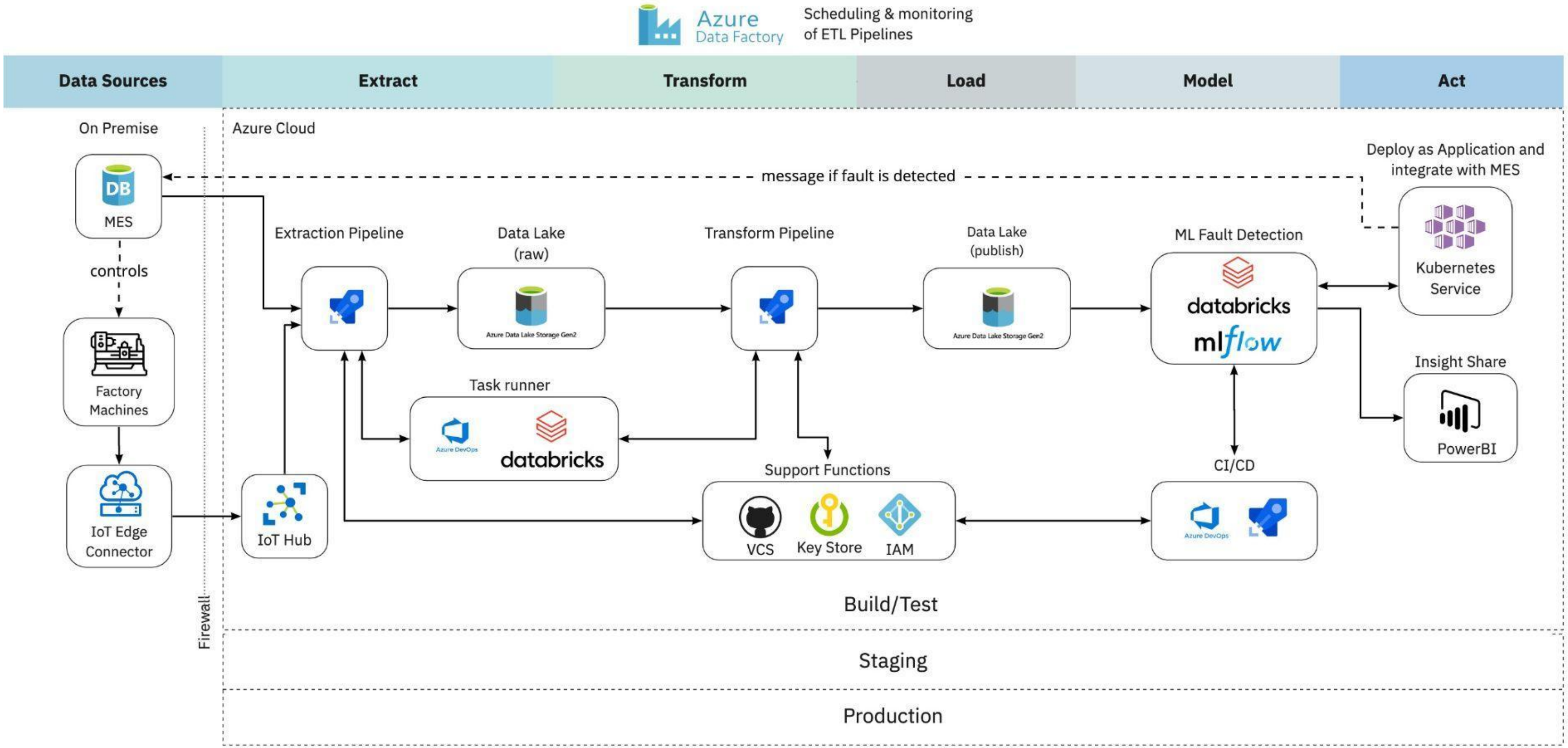
Benefits

- **Improve quality control**
- **Increase customer satisfaction**
- **Achieve smarter data-driven decisions**
- **Reduce rework**
- **Enable continuous data flow**
- **Enable scalability of detection systems**

XYZ Products

Implementation

… at scale enabled by future-proof Cloud Architecture



XYZ Products

Scalable cloud architecture brings data-driven insights into production

Data Source

Production Data
MES[1] captures data from the production process like steel dimensions, types and characteristics

Machine Sensors
Provide additional data that are made available directly through the IoT Edge Connector

ETL

Extraction pipeline for moving the data from the MES and IoT Hub into the cloud data lake

Transformation logic for scalable pipelines with version control support

Loading of data into the data lake to be used in analytics and model building

Model

Databricks
Scalable ML environment with ML Flow experiment tracking and model registry

CI/CD Integration
Continuous adding of ideas into production, maintained in releasable state, supported by model governance

Act

Production Integration
Integrate with production systems to notify of detected faults and minimise downtime and costs

Analytics
Access to centralised data builds up internal analytics capabilities to improve the manufacturing process

[1]MES = Manufacturing Execution System

Cost Insights

Best practices for cost-effective cloud migration to maximize and sustain profits

Costs

Running cloud costs - analyse direct and indirect cost obligations in the current IT setup

Investments - connecting manufacturing equipment to Azure through the IoT Edge Connector

Operating expenditures - tech onboarding, licenses, training, IT facility and maintenance costs

Cost-effective recommendations

Pre-migration planning - forecasting and plan for future resource requirements

Cloud Computing Cost Optimization - set up billing alerts and budget during migration process. Set up proper mechanisms to optimise usage of data lakes

Balance Cost VS Risk- considerations of security and compliance

With predictive maintenance in place, it is possible to achieve OEE above 85%, constituting to world class industry level

Automating quality testing using machine learning can increase defect detection rates up to 90%

Migration to the cloud creates foundation for end-to-end digital transformation

Why?

Cost-reduction by cause of reduced downtime and personnel costs through detecting the faults earlier in the production process

Improved Investments in the Supply Chain by building knowledge about fault types and their most significant cost drivers

How?

AI-powered fault detection model enabling fault insights

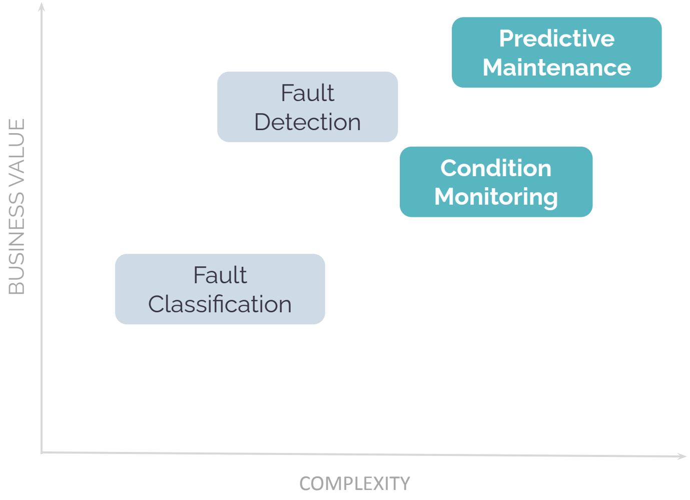
Cloud Architecture empowering fault detection and a continuous insights flow

Why Now?

Unlock the potential of data-driven value creation to reduce costs in the short-term and build long-term capabilities

What is Next?

Further leverage the Cloud Architecture by enabling discovery of machine anomalies before they become critical issues to ensure maximum Overall Equipment Effectiveness (OEE) with rich insights and automatic alerts



Appendix

Timeline

Delivering ETL solution within 3 months

01

W2 W3 W4 W5

- Assess the current manual process & fault pattern analysis
- Gap analysis
- Establish to-be state
- Evaluate solution complexity/value

02

W6 W7 W8 W9

- Set up IoT edge connector and image recognition
- Set up and schedule ETL pipes
- Develop ML models
- Set up CI/CD practices
- Build test environment

03

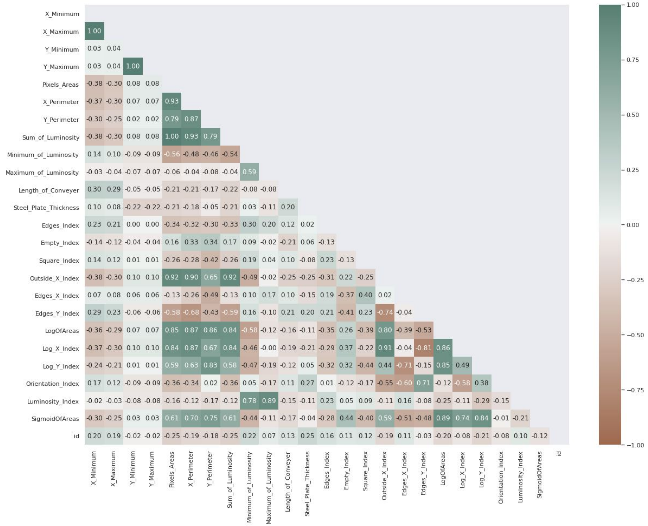
W10 W11 W12 W13

- Execute in staging and production environment

XYZ Products

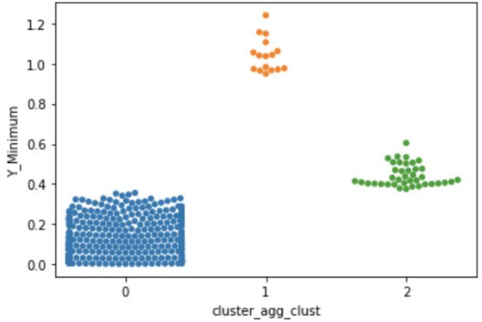
Data Insights

Analysing data patterns

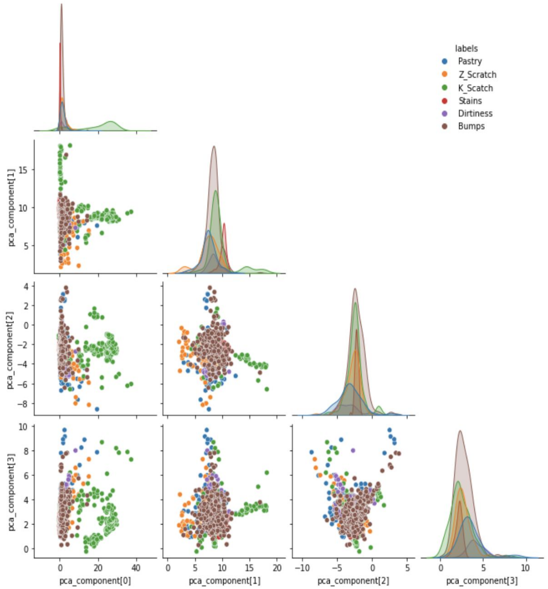


Some features depict strong correlations suggesting potential for reducing dimensionality and therefore complexity of models

Replace Y_Maximum and Y_minimum with Y_minimum **eliminating Y_maximum**
Replace X_Maximum and X_minimum with X_minimum **eliminating X_Maximum**
Sum_ofLuminiosity and Pixels_Area with Sum_ofLuminiosity **eliminating Pixels_Area**



Clustering was performed on “Other Faults” to identify potential for subgroups, 3 potential clusters were detected (Y-Dimensions)



PCA shows that K_scratch fault type is quite distinct

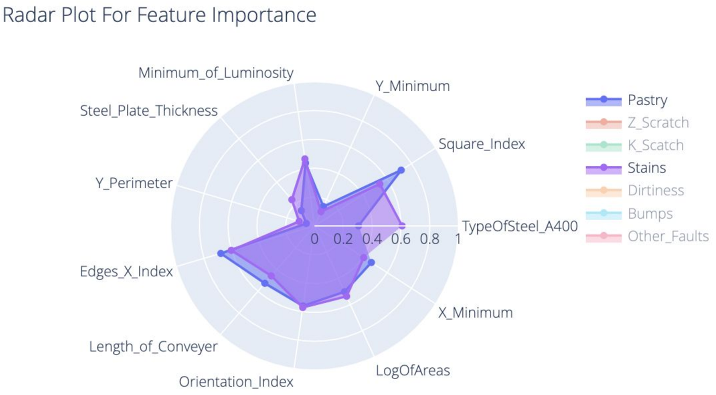
Other identified faults might have some degree of overlap

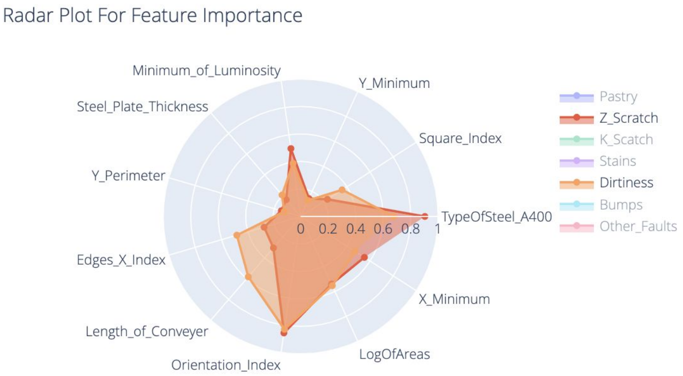
Unidentified faults could be over-labelling

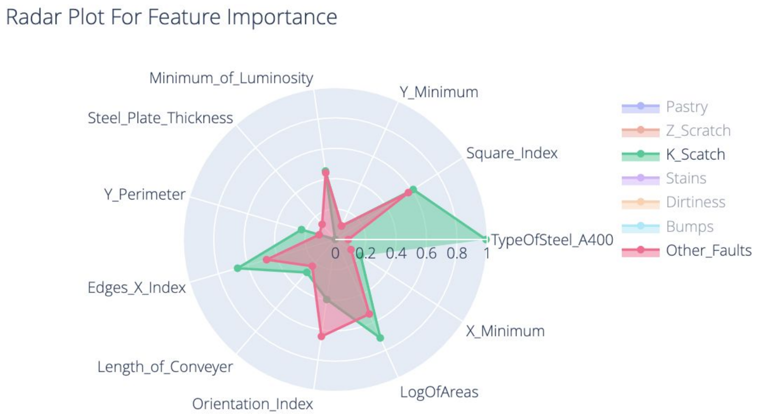
XYZ Products

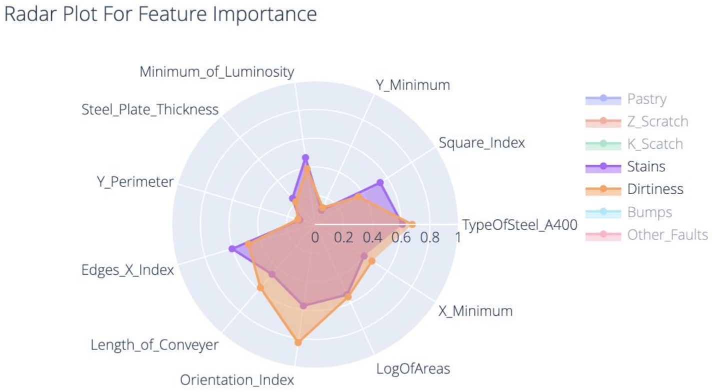
Data Insights

Certain degree of label overlap is observed



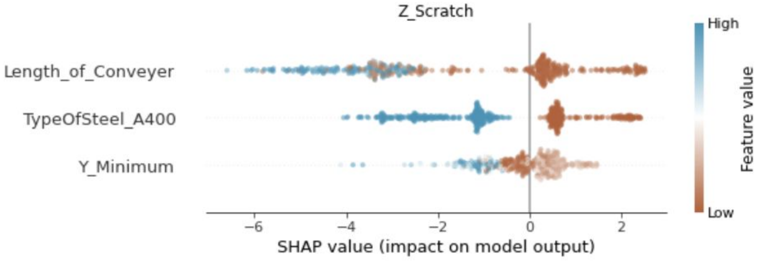


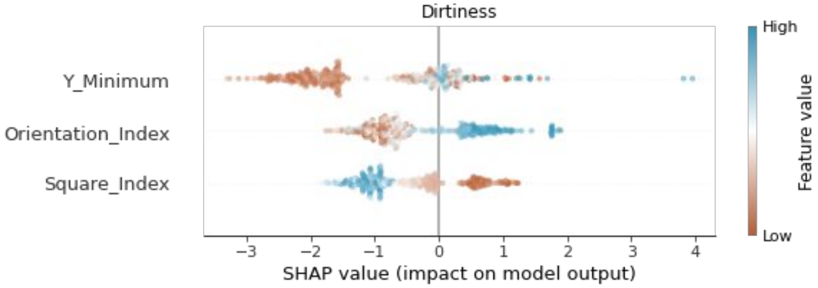


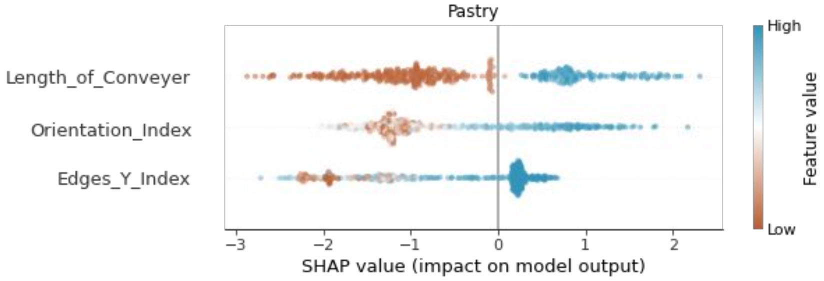


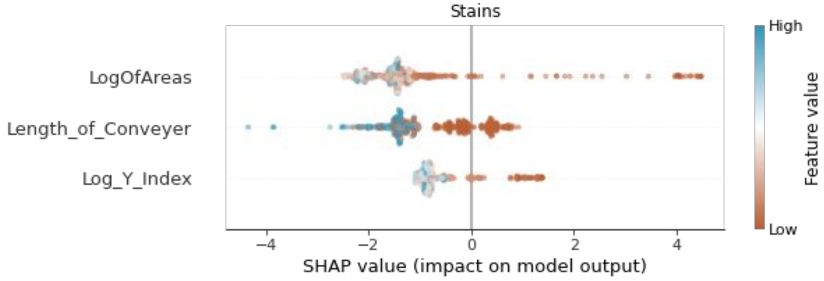
Products

Feature Impact of less common faults



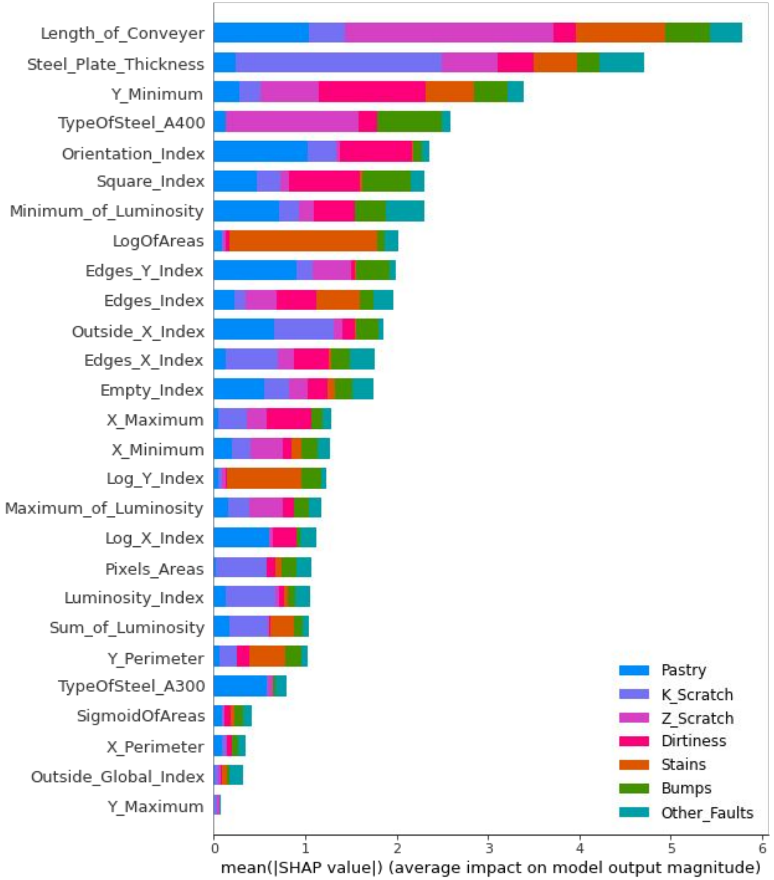




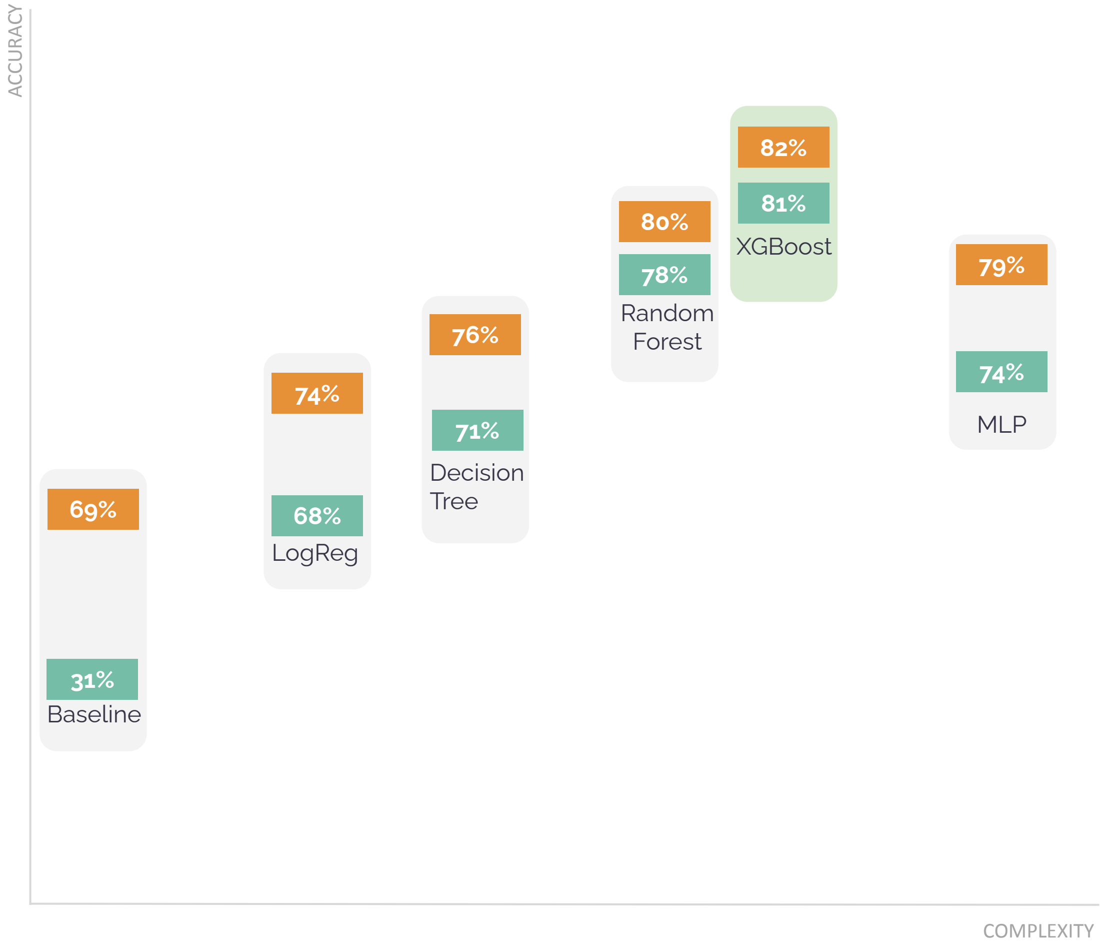


Data Insights

Overall Feature Importance



XYZ Products



Model Selection

Achieving over 80% accuracy

ML algorithms tuned with **hyperparameters** were run to identify best performing models, **cross validation** was used in each

- **Oversampling** was implemented to avoid incorporating bias towards majority class, resulting in higher accuracy
- Experiments with and without **outliers** were run, keeping non-extreme outliers rendered better results
- **Feature selection** was implemented, strongly correlated features were dropped reducing model complexity
- **Robust scaling** was used to facilitate feature interpretation by distance-sensitive models

Binary Classification

"Other faults" being the majority class suggests potential for binary classification

Identified labels were grouped into "common" due to possible overlaps

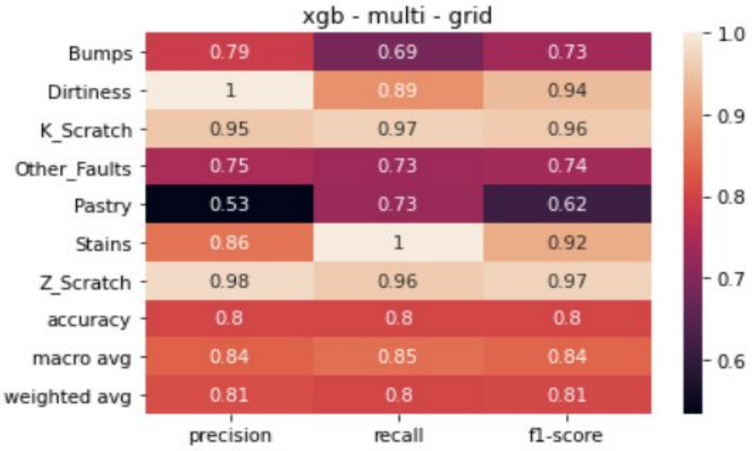
Multi-Class Classification

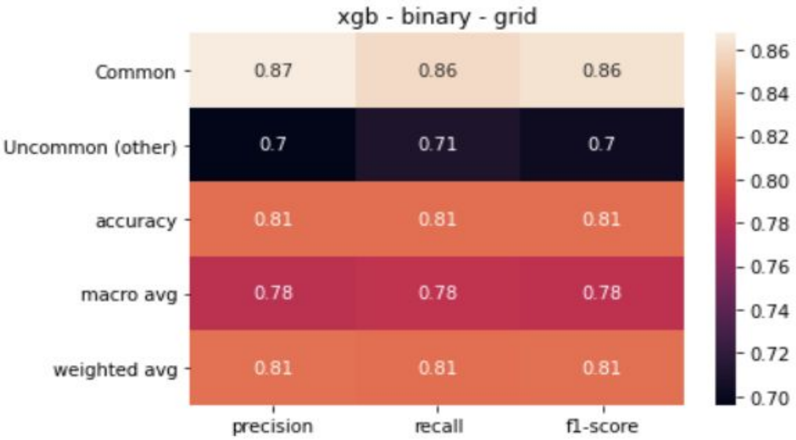
7 label classification performs only slightly worse than binary suggesting stronger class separation patterns and potential subclusters to be identified in "Other" faults

XYZ Products

Results

XGBoost classification matrices





XYZ Products

Process Delta

Automating the Production Process

Fault Classification Today

Customer order

Incoming steel plate orders

Production line

Faulty steel plates returned into relative production line stages after customer complaints

Shipment

Shipping non-quality assured steel plates

Manual Fault Classification

Faulty steel plates are returned by customers and manually classified

Automated Fault Detection and Prevention

Model Feedback

Model Feedback

Fault Prevention

Customer order

Incoming steel plate orders from satisfied customers

Production line

Faulty steel plates returned into relative production line stages after fault detection

Automated Fault detection

Replace manual labelling with automatic fault detection

AI and optical recognition of plates

Dynamic flow of insights back into production line allows for preventive maintenance and data-driven decisions

Shipment

Shipping quality-assured steel plates results in happier customers, improved reputation and reduced costs

Fault Classification /Insights

Faulty steel plates are returned by customers and utilized to enhance the model

XYZ Products