# Automated Inspection System Alert Project

## Introduction

North American Stainless (NAS) is a producer of a range of goods using distinct machine process lines. Many automatic visual inspection systems, like the Parsytec system, find possible flaws in the items throughout the production process. In addition, human inspectors confirm the flaws detected by the automatic system and find any flaws that might have gone unnoticed. Customers also report products that were defective but were missed in the production process.

The goal of NAS's Automated Inspection System Alert Project is to improve the company's production workflow's fault detection and classification procedures. The project uses data aggregation from multiple sources, such as automatic vision inspection systems, human inspectors, and customer claims, to increase operational efficiency and product quality. This report describes the methodology, the preparation stages for the data, the classification strategies used, and the analysis's findings.

## Data Set Description

The dataset comprises 13 datasets provided by NAS's manufacturing and inspection systems. These datasets contain information about various aspects of the manufacturing process, including process details, product specifications, inspection data, defect codes, and coil information. The data is generated from the company's production systems, inspection systems, and customer feedback systems.

This dataset has been used for the following research and analysis:

● Aggregating data from different sources to obtain a comprehensive view of the product's production process.

● Comparing and classifying whether human inspectors agreed or disagreed with the automatic inspection system for various defect codes.

● Verifying if customers claimed defects that were not caught during the production process.

● Building algorithms to alert operators about potential defects based on the aggregated data.

The dataset contains various characteristics and features related to the manufacturing process, product specifications, inspection data, defect codes, and coil information. Some of the key characteristics include:

● Process details: ProcessID, ProductID, ProductDivision, CrewID, ShiftID, LineID, StartDate, EndTime, etc.

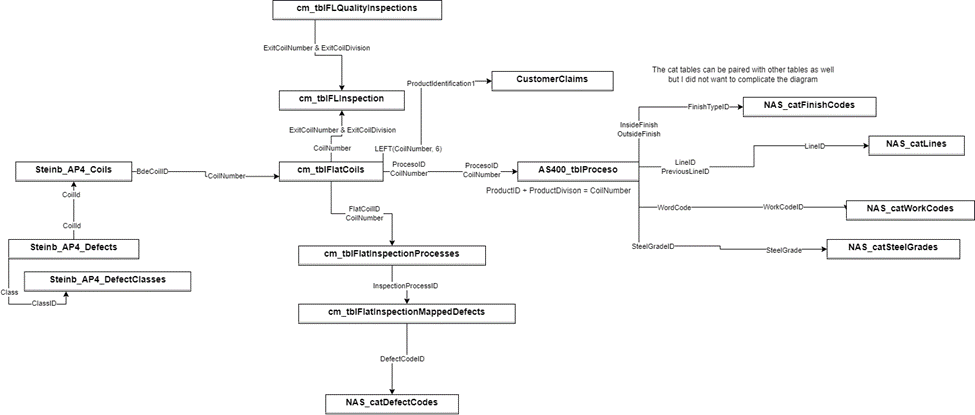
● Product specifications: SteelGradeID, CurrentGuage, CurrentWidth, SheetLength, CoilLength, NetWeight, etc.

● Inspection data: InspectionDate, InspectionTime, Percent1AQualityExt, Percent1BQualityExt, Percent2QualityExt, PercentScrapQualityExt, MnDefect1, MnDefect2, etc.

● Defect codes: DefectCodeID, SideID, FaceID, StartPosition, Length, QualityID, DefectCount, etc.

● Coil information: CoilId, StartTime, EndTime, Grade, Length, Width, Thickness, Weight, Charge, etc.

Diagram:



## Research Questions

The main classification questions of interest in this project were:

● Classify whether the human inspectors agreed or disagreed with the customer claim defect identification.

● Classify whether a defect claimed by a customer was caught or missed during the production process.

● Predict the length of defect coils

● Predict the net weight of defect coils

● Predict the number of defect count

● Find potential relations between defect codes

## Methodology

## Data Preprocessing Techniques

Before performing any analysis or building models, several data preprocessing tasks were carried out on the dataset. These tasks were essential to ensure the data was clean, consistent, and ready for analysis. Here are the preprocessing steps performed:

* Handling missing values: Missing values were identified in the dataset, and appropriate techniques were applied to handle them. These techniques included:
* Imputation: Missing numerical values were imputed using techniques like mean/median imputation or more advanced methods like k-Nearest Neighbors imputation, depending on the distribution of the data and the pattern of missingness.
* Dropping rows/columns: In cases where the missing values were significant or imputation was not feasible, rows or columns with missing values were dropped.
* Data type conversion: The data types of various features were checked and converted to appropriate types (e.g., converting string dates to datetime format) to ensure consistency and enable efficient processing.
* Handling outliers: Outliers in numerical features were identified and addressed using techniques like winsorization or capping, depending on the nature of the outliers and the distribution of the data.
* Feature encoding: Categorical features were encoded using techniques like one-hot encoding or label encoding, depending on the nature of the categorical data and the requirements of the analysis or modeling techniques.
* Feature scaling: Numerical features with different scales were scaled using techniques like StandardScaler or MinMaxScaler to ensure that features with larger scales did not dominate the analysis or modeling process.
* Feature selection: Irrelevant or redundant features were identified and removed from the dataset to improve the efficiency and performance of the analysis or modeling techniques. Feature selection techniques like correlation analysis, recursive feature elimination, or regularization methods (e.g., Lasso, Ridge) were applied.
* Data integration: Since the dataset comprised multiple datasets, data integration techniques were applied to combine the relevant features from different datasets into a single dataset for analysis.

The choice of preprocessing techniques was based on the characteristics of the data, the distribution of the features, and the requirements of the analysis or modeling techniques. For example, imputation techniques were chosen based on the pattern of missingness and the distribution of the data, while feature selection techniques were chosen based on the correlation between features and the presence of redundant features.

## Modeling Techniques

### Technique 1

### Technique 2

### Technique 3

In order to predict the length of the defect codes, information from the “FlatInspectionMappedDefects” table was merged with “FlatInspectionProcesses”, “DefectCodes”, and “FlatCoils”. To answer this question, the results from four modeling techniques were utilized and compared. These techniques were:

**Logistic Regression:** A process that models the probability of an outcome given the series of input variables from this dataset.

**Decision Trees:** An algorithm that creates a hierarchical, tree structure which contains a series of nodes or decisions in order to classify the defect lengths into categories.

**Naive Bayes:** An advanced classification technique based on Bayes theorem that classified defect lengths based on input features from this dataset, while assuming independence between these features.

**K-Nearest Neighbor:** An algorithm that utilizes the proximity between instances to predictions about the classification or grouping of a point, specifically the defect length.

Both 5-fold and 10-fold cross-validation was performed, and the resulting mean cross-validation scores and their standard deviations were used to determine the best model for answering this research question.

### Technique 4

In order to predict the net weight of the defect coils, information from the “Proceso” table was utilized. To answer this question, the results from four modeling techniques were compared. These techniques were:

**Logistic Regression:** A process that models the probability of an outcome given the series of input variables from this dataset.

**Decision Trees:** An algorithm that creates a hierarchical, tree structure which contains a series of nodes or decisions in order to classify the net weight into categories.

**Naive Bayes:** An advanced classification technique based on Bayes theorem that classified the net weight based on input features from this table, while assuming independence between these features.

**K-Nearest Neighbor:** An algorithm that utilizes the proximity between instances to predictions about the classification or grouping of a point, specifically the net weight.

Both 5-fold and 10-fold cross-validation was performed, and the resulting mean cross-validation scores and their standard deviations were used to determine the best model for answering this research question.

### Technique 5

**Logistic Regression:**

* + Mean CV Score: The mean cross-validation score across all folds is approximately 0.908.
  + Standard Deviation of CV Scores: The standard deviation of cross-validation scores is approximately 0.020.
  + Observation: The logistic regression model achieves a relatively high mean cross-validation score, indicating that it performs reasonably well in predicting total defect counts based on defect codes.

**Decision Tree:**

* + Mean CV Score: The mean cross-validation score across all folds is approximately 0.908.
  + Standard Deviation of CV Scores: The standard deviation of cross-validation scores is approximately 0.020.
  + Observation: Similar to logistic regression, the decision tree model also achieves a high mean cross-validation score, suggesting good performance in predicting total defect counts.

**Naive Bayes:**

* + Mean CV Score: The mean cross-validation score across all folds is approximately 0.062.
  + Standard Deviation of CV Scores: The standard deviation of cross-validation scores is approximately 0.022.
  + Observation: The naive Bayes classifier yields a much lower mean cross-validation score compared to logistic regression and decision tree models, indicating poorer performance in predicting total defect counts based on defect codes.

**k-Nearest Neighbors:**

* + Mean CV Score: The mean cross-validation score across all folds is approximately 0.909.
  + Standard Deviation of CV Scores: The standard deviation of cross-validation scores is approximately 0.020.
  + Observation: The k-nearest neighbors classifier achieves a similar mean cross-validation score to logistic regression and decision tree models, suggesting comparable performance in predicting total defect counts.

**Analysis:**

* + Logistic regression, decision tree, and k-nearest neighbors classifiers exhibit relatively high mean cross-validation scores, indicating good performance in predicting total defect counts.
  + Naive Bayes, however, performs significantly worse compared to the other classifiers, as evidenced by its much lower mean cross-validation score.

### Technique 6

**Apriori Algorithm**

The Apriori algorithm is a classical algorithm in data mining used for association rule mining. It's designed to find frequent itemsets in transaction databases and generate association rules.

* + Support: This indicates the frequency of occurrence of the itemset in the dataset. ` Higher support values indicate that the itemset occurs frequently in the dataset.
  + Confidence: Confidence measures the reliability of the rule. It is the likelihood of the consequent (output) occurring given that the antecedent (input) has occurred. Higher confidence values indicate a stronger association between the antecedent and consequent.
  + Lift: Lift measures how much more likely the consequent is given the antecedent compared to its prior probability. A lift greater than 1 indicates that the antecedent and consequent appear together more often than expected by chance. Lift values closer to or greater than 1 indicate stronger associations.

### Technique 7

## Results

Question 1 Results: Classify whether the human inspectors agreed or disagreed with the customer claim defect identification.

| **Modeling Technique** | **Accuracy** | **Mean CV Score** | **Standard Deviation** |
| --- | --- | --- | --- |
| Logistic Regression |  |  |  |
| Naive Bayes |  |  |  |
| K-Nearest Neighbor (KNN) |  |  |  |

Question 2 Results: Classify whether a defect claimed by a customer was caught or missed during the production process.

| **Modeling Technique** | **Accuracy** | **Mean CV Score** | **Standard Deviation** |
| --- | --- | --- | --- |
| Logistic Regression |  |  |  |
| Naive Bayes |  |  |  |
| K-Nearest Neighbor (KNN) |  |  |  |

Question 3 Results: Predict the length of defect coils

| **Modeling Technique** | **Mean CV Score** | **Standard Deviation** |
| --- | --- | --- |
| Logistic Regression | 0.63823 - 5 Folds | 0.02208 - 5 Folds |
| Decision Tree | 0.89934 - 5 Folds | 0.00349 - 5 Folds |
| Naive Bayes | 0.67901 - 5 Folds | 0.01562 - 5 Folds |
| K-Nearest Neighbors (KNN) | 0.66781 - 5 Folds | 0.02464 - 5 Folds |

Conclusion: Based on the mean cross-validation scores and standard deviations, the decision tree model appears to have the highest average performance for predictions when compared to the other models included. However, it's essential to consider other factors such as model complexity, interpretability, and computational efficiency when choosing the final model.

Question 4 Results: Predict the net weight of defect coils

| **Modeling Technique** | **Mean CV Score** | **Standard Deviation** |
| --- | --- | --- |
| Logistic Regression | 0.92775 - 10 Folds | 0.00677 - 10 Folds |
| Decision Tree | 0.98899 - 5 Folds | 0.00126 - 5 Folds |
| Naive Bayes | 0.91696 - 10 Folds | 0.01101 - 10 Folds |
| K-Nearest Neighbors (KNN) | 0.37241 - 5 Folds | 0.16553 - 5 Folds |

Conclusion: Based on the mean cross-validation scores and standard deviations, the decision tree model appears to have the highest average performance for predictions when compared to the other models included. However, the logistic regression and naive bayes models both also result in high mean cross-validation scores (over 0.90), and therefore create high-performing predictions.

Question 5: Predict the number of defect count

| **Modeling Technique** | **Mean CV Score** | **Standard Deviation** |
| --- | --- | --- |
| Logistic Regression | 0.9084 - 5 folds  0.90888888 - 10 folds | 0.01958 - 5 folds  0.030550 - 10 folds |
| Decision Tree | 0.90842105 - 5 folds  0.9088888 - 10 folds | 0.019580- 5 folds  0.03055050 - 10 folds |
| Naive Bayes | 0.06157894 - 5 folds  0.061111 - 10 folds | 0.0218658 - 5 folds  0.05 - 10 folds |
| K-Nearest Neighbors (KNN) | 0.90842105 - 5 folds  0.90888888- 10 folds | 0.019580079 - 5 folds  0.0305505 - 10 folds |

Question 6: Find potential relations between defect codes

Some potential relations between defect codes:

**defectcode\_C21 -> defectcode\_C23**

Confidence: 36.36% - 50%

Lift: 3.90 - 105.69

Indicates a moderate to strong association between defect codes C21 and C23. The lift values suggest that this association is significant.

**defectcode\_C21 -> defectcode\_C47**

Confidence: 33.33% - 100%

Lift: 3.58 - 8.85

Suggests a moderate to strong association between defect codes C21 and C47. The high lift values indicate a significant relationship.

**defectcode\_C42 -> defectcode\_C23**

Confidence: 22.58%

Lift: 69.43

Indicates a moderate association between defect codes C42 and C23.

**defectcode\_C47 -> defectcode\_C23**

Confidence: 25.93%

Lift: 79.71

Shows a moderate association between defect codes C47 and C23.

**defectcode\_C42 -> defectcode\_C47**

Confidence: 50%

Lift: 135.28

Suggests a strong association between defect codes C42 and C47.

**defectcode\_C42 -> defectcode\_D16**

Confidence: 22.58%

Lift: 39.16 - 50.20

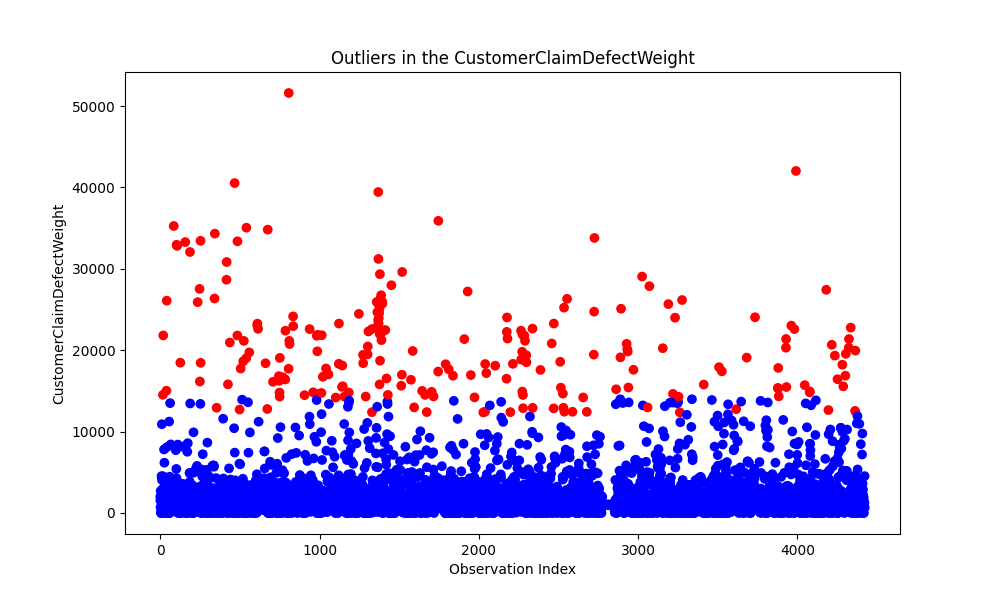
Indicates a moderate association between defect codes C42 and D16.

**defectcode\_C47 -> defectcode\_D16**

Confidence: 21.43% - 25%

Lift: 7.51 - 13.21

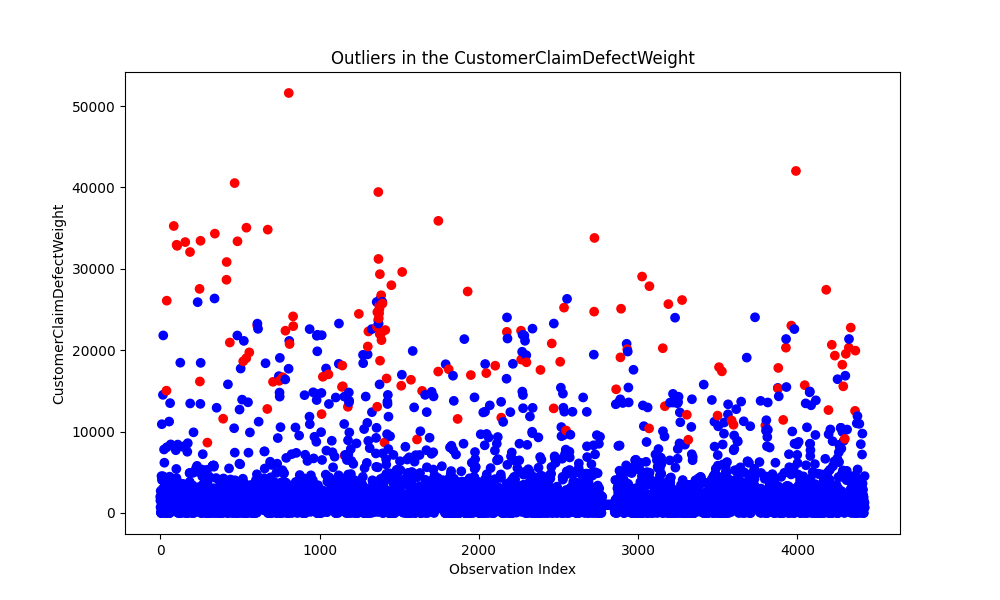
Shows a moderate association between defect codes C47 and D16.

Anomaly Detection: Customer Claims Data

**K-Nearest Neighbor**

* The blue data points represent the inliers or normal observations, while the red data points are identified as outliers by the kNN algorithm.
* The distribution of the outliers (red data points) appears to be relatively uniform, with no obvious patterns or clusters, which is consistent with the nature of outliers being rare and isolated occurrences.
* The number of outliers detected by the kNN method seems to be significant, as indicated by the relatively large number of red data points compared to the blue inliers.

The kNN approach appears to have successfully identified a substantial number of outliers in the "CustomerClaimDefectWeight" variable across the entire dataset, with a higher concentration of outliers towards the higher end of the "Observation Index" range. This information can be valuable for further analysis and data cleaning for handling these outliers.



**Isolation Forest**

* The majority of the points are concentrated in the lower region of the plot, forming a dense cluster of blue points. However, there are several red points scattered above this cluster, which are likely the identified outliers.
* The presence of outliers is particularly noticeable at higher values of the "Observation Index." There are several isolated red points at the top of the plot, indicating significantly higher "CustomerClaimDefectWeight" values compared to the majority of the data points.

The Isolation Forest algorithm is a powerful and efficient technique for detecting outliers in datasets without requiring labeled data, and it is particularly useful in scenarios where anomalies or outliers are of interest or need to be identified and analyzed further.

## Conclusion

In this project, various tasks were performed to address the classification problems related to defect detection in the manufacturing process. The tasks included:

● Data preprocessing: Handling missing values, data type conversion, outlier treatment, feature encoding, feature scaling, feature selection, and data integration.

● Feature engineering: Deriving additional features to improve the performance of classification models.

● Classification modeling: Applying various classification algorithms, including logistic regression, decision trees, random forests, gradient boosting machines, support vector machines, and neural networks.

● Model evaluation: Assessing the performance of the classification models using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC, with cross-validation techniques (5-fold and 10-fold) to ensure robust evaluation and prevent overfitting.

● Model selection: Selecting the best-performing models based on the evaluation metrics and the specific requirements of the classification tasks.

Throughout the project, several lessons were learned:

● Data preprocessing is crucial: Properly handling missing values, outliers, and data inconsistencies is essential for obtaining reliable and accurate results from any machine learning or data analysis task.

● Feature engineering can significantly improve model performance: Deriving relevant features from the existing data can provide valuable insights and enhance the predictive power of the models.

● Domain knowledge is valuable: Incorporating domain knowledge and insights from subject matter experts can help in interpreting the results, identifying relevant features, and understanding the practical implications of the models.

While the project aimed to address the classification tasks related to defect detection, there are still some outstanding questions and potential areas for further investigation:

● How can the models be deployed and integrated into the existing production systems to provide real-time alerts and support decision-making processes?

● Are there any other sources of data, potentially data pertaining to the makeup of the material and/or the chemical information, that could be incorporated to further improve the accuracy and robustness of the models?

These outstanding questions could be explored in future projects or as extensions of the current work to further enhance the defect detection and monitoring capabilities of the manufacturing process.

## Reflection

If I had the chance to take the Data Mining course again, I would prioritize strong foundational knowledge, effective time management, hands-on practical experience, collaborative learning, staying updated with the latest developments, and applying the concepts to real-world problems. By following these recommendations, students can maximize their learning experience and better prepare themselves for success in the field of data mining.