Identifying the Type of Finished Product Based on Product Elements Using Machine Learning

A Partnership Project

with

Ernst and Young (EY)

Sujatha Ramesh

Matric Number: 30006034

10th June 2022

Table of Contents

[Executive Summary 3](#_Toc105074702)

[1 Introduction 4](#_Toc105074703)

[1.1 Motivation 5](#_Toc105074704)

[1.2 Business Problem 5](#_Toc105074705)

[2 Methodology 6](#_Toc105074706)

[2.1 Understanding the Business Problem 6](#_Toc105074707)

[2.2 Data Collection 6](#_Toc105074708)

[2.3 Data Cleaning and Data Pre-Processing 7](#_Toc105074709)

[2.4 Feature Engineering 9](#_Toc105074710)

[2.5 Model Selection 12](#_Toc105074711)

[3 Results 13](#_Toc105074712)

[4 Conclusion and Future Work 14](#_Toc105074713)

[5 References 15](#_Toc105074714)

# Executive Summary

The goal of this project is to use machine learning approaches to assign the correct product to an incomplete Bill of Material using the product elements. The dataset used for the project consisted of 341 samples of different Bill of Materials containing subcomponents and their respective finished products from a dairy manufacturing company. Several data pre-processing and feature engineering steps were performed on the dataset to ensure that the chosen algorithm performs to its optimum capability. An algorithm was established which helped to select all finished products that contained all product elements in their construct from a given list of subcomponents in a bill of material. This brings out all the likely products that the given Bill of Material component list could represent. Hence, the next step of the project was to optimize and determine which is the likely product? From understanding of the business problem, it was determined that one of the best models that gives an optimal solution would be the Naive Bayes Algorithm. After the Naive Bayes Algorithm was implemented, the most likely finished product from the given list of components was decided and obtained.

# Introduction

A Bill of Materials (BOM) is a list of all necessary materials and components required to make a product. The data structure of each product in the factory can be described by the BOM. It describes the parent–child relationship of the items required for a specific item. It includes not only the fundamental properties of materials, suppliers, processes, and so on, but also the number of components required for each part of the relationship. The BOM is a data structure that is shaped like a tree. The BOM in the factory connects the entire factory product's life cycle. It usually appears in a hierarchical format, with the highest level displaying the finished product and the bottom level showing individual components and materials (Zhou & Cao, 2019) .

Diagram

Description automatically generated

Figure 1: Sample of Bill of Material Structure (OpenBOM (openbom.com), 2018)

Manufacturing bills of materials and engineering bills of materials are the two main types of BOMs.  A manufacturing bill of materials lists all the assemblies and parts that go into making a final product that can be shipped. It also includes the materials needed to transport the goods to the client in its packaging. It contains processes that must be implemented on the product before it can be finished, as well as all the information needed for manufacturing activities. The design of the finished product is defined by an engineering bill of materials. All alternative and substitute part numbers and parts in the drawing notes are included. The product code, part name, part number, part revision, description, quantity, unit of measure, size, length, weight, and specifications or characteristics of the product are all listed on each line of the bill of materials (Grant, 2022).

## Motivation

Any company that wishes to take its manufactured items from concept to customer with ease, starts with a detailed manufacturing Bill of Materials (BoM). Assume a company enters production with a bill of materials that is incorrect or missing critical information or instructions. The production work could cause significant delays or perhaps halt the entire manufacturing process. A poorly prepared Bill of Materials can be extremely costly to a company, in terms of time delays, capital and human resources (Dear Systems, 2021) .

As a result, BOM Analytics is critical to the manufacturing industry for a variety of reasons:

* Proactively manage component obsolescence and compliance.
* Avoid essential component shortages to fulfill production and sales targets.
* Eliminate redundant components, suppliers, and surplus inventories.
* Avoid missed opportunities in areas where environmental compliance laws apply.
* Manage extended supply chains and eliminate counterfeit part concerns.
* Identify technology advances that can alter designs, sources, or margins across the product cycle.

## Business Problem

How can we identify the type of the finished product based on the product elements, using Machine Learning?

# Methodology

## Understanding the Business Problem

As the risks of obsolescence, counterfeiting, and non-compliance continue to rise, managing components and suppliers is more important than ever. A successful BOM analysis system will enable businesses to accelerate the introduction of new products, minimize costly production disruptions or product redesigns, and improve product sustainability over longer service lives (IHS Markit, 2016).

In detail, many of the finished products in advance manufacturing sector have lot of sub-components that are commonly shared.​ So, in approaching to solve this problem, we defined a clear goal which was to correctly identify the finished product from bill of material containing different components and product elements.

## Data Collection

The data used for this analysis was an extract collected in an excel format from a dairy manufacturing company, a client of Ernst and Young (EY). BoM data is classified into two types: Single level and Multi level. In this project, we had access to a single level BoM data with only two layers – End Product and subcomponents.

Diagram

Description automatically generated

Figure 2: Single Level and Multi Level BOM

The dataset had a total of 341 data samples which contained the following variables:

* **STLNR**: refers to BoM unique ID
* **MATNR**: Header of the BoM (Parent article)
* **MAKTX**: Header description (Name of the article)
* **MTART**: Material Type (KMAT refers to configured material or produced finished good)
* **MATKL**: Material class (category)
* **GROES**: Material Size
* **PRDHA**: Product Structure Code (unique code, referring to the product structure classes)
* **MEINS**: Base unit of measure (ST refers to piece)
* **IDNRK**: BoM Item (Child article)
* **STPO\_MTART**: Material type of BoM Item
* **STPO\_MATKL**: Material class of BoM Item
* **STOP\_PRDHA**: Product Structure Code of BoM Item
* **STOP\_MAKTX**: BoM Item Description.

## Data Cleaning and Data Pre-Processing

The transformation of the raw dataset into a comprehensible format is known as data preprocessing. Data preprocessing is a critical step in data mining that improves data efficiency. Data preprocessing procedures have a direct impact on the results of any analytic algorithm. All pre-processing steps were implemented in python with the following steps:

* Imported the dataset and libraries needed for the analysis

Text

Description automatically generated

Figure 3: Import Libraries

Graphical user interface, application

Description automatically generated with medium confidence

Figure 4: Import Dataset

* **Dealing with Missing Values and Datatypes**: As a next step, this project evaluated the missing values within the data set to better understand how to deal with them. First, we search for patterns within the variables through visual inspection. There, we speculate how the distribution of the missing values would look like if they were available by discussing if the missing data is “missing completely at random”, “missing at random” or “not missing at random.” From understanding the business problem, we decided to drop the rows with NA values. Also, data types are checked, to ensure that accuracy in the analysis. The fields were changed to strings and unnecessary preceeding zeros were removed from the IDNRK, MATNR and STPO\_PRDHA columns.

Text

Description automatically generated with medium confidence

Figure 5: Data Cleaning Steps

## Feature Engineering

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data (Browniee, 2014).

In this project, we had to restructure our data to accurately address our business problem. Here, are the following feature engineering steps that were done in this project.

* **Feature Selection**: From the understanding of the problem and acquiring necessary domain knowledge, we selected only to relevant features to conduct our analysis - STNLR, MATNR & IDNRK.

Graphical user interface, table

Description automatically generated

Figure 6: Features Selected for Analysis

* **Created New Feature (IDRNK\_Count)**: Next, we derived a new feature from the IDNRK\_Count which is the frequency of occurrence of the subcomponents (IDNRK) in a finished product for a specific BOM. This is important as it is needed to solve the business problem.

Table

Description automatically generated

Figure 7: New Feature IDNRK\_count added to the dataset

* **Logic to get all products from a given subcomponent list**: In identifying the type of the finished product, a logic was created which helps select all products that have all the given subcomponents in its product construct.

For example, here is a given list of subcomponents which is the input.

A picture containing calendar

Description automatically generated

Figure 8: Sample of a given component list

The logic then brings as an output all the products that have these subcomponents in their product construct.

Table

Description automatically generated

Figure 9: Three Products selected by the logic

This has solved the first part of the business problem as the logic shows all product have a given list of subcomponents. The next question is how to determine the most likely finished product out of the selected products?

* **Created a New Feature (STNLR\_MATNR)**: This feature was derived from the concatenation of the STNLR (BOM Unique ID) and MATNR (Parent/Finished Product) features to allow us to derive a unique identifier for each bill of material and its corresponding finished product. This was also important to ensure accurate frequency of occurrence in subcomponents for each specific bill of material instance.

Table

Description automatically generated

Figure 10: New feature derived STNLR\_MATNR

* **Data Transformation:** To ensure that the data better represent the underlying problem to the predictive model, the data was transformed. The subcomponents (IDNRK) became the features, their respective frequencies (IDNRK\_count) as rows in the dataset with the new feature STNLR\_MATNR becoming the target feature to be predicted. The test component list was transformed to this data structure. The data transformation steps were also implemented on the test component list set.
* **Dealing with categorical variables:** The STNLR\_MATNR categorical variable representing the target feature to be prediced was modified by using label encoding to represent the different finished products. This becomes are training data for the machine learning model.

Graphical user interface, application

Description automatically generated

Figure 11: Transformed data to feed into the predictive model

## Model Selection

The next step was to decide which model would predict the most likely product. From our understanding, this is an optimization problem. Hence, we decided to go with the **Naïve Bayes Model**. A Naïve Bayesclassifier is a probabilistic machine learning model that’s used for classification task. The Bayes theorem lies at the core of the classifier.

Bayes Theorem:

Text, letter

Description automatically generated

Figure 12: Bayes Theorem Formula

*P*(*c|x*) is the [posterior probability](https://en.wikipedia.org/wiki/Posterior_probability) of *class* c given *predictor* (*features*).

*P*(*c*) is the probability of *class*.

*P*(*x|c*) is the [likelihood](https://stats.stackexchange.com/questions/314623/naive-bayes-likelihood) which is the probability of *predictor* given *class*.

*P*(*x*) is the [prior probability](https://en.wikipedia.org/wiki/Prior_probability) of *predictor*.

We can calculate the probability of***c*** occurring if ***x*** has already occurred using Bayes' theorem. The evidence is ***x***, and the hypothesis is ***c***. The predictors/features are assumed to be independent in this case. That is, the presence of one feature has does not affect the other. As a result, it is said to be naïve.

The Naive Bayes classifier has the advantage of being simple and quick to predict the test data set's class. It's also good at multi-class prediction. A Naive Bayes classifier performs well when the premise of independence stays true, and there is less training data. The Naive Bayes classifier is a good approach for this project as we a small training data.

# Results

The Naive Bayes Classifer was trained using the training data and then deployed on the test data which is the given component list.

Graphical user interface, text, application

Description automatically generated

Figure 13: Implementation of Naive Bayes Model

After the model was completed, the outcome of the probability distribution score between the three products were displayed. In this case, based on the higher probability (0.33701651), the predicted product can be assumed to be **[0]** i.e**.,** STNLR is **364014\_79101000040**.

# Conclusion and Future Work

From the outcome of this project, the finished product in a dairy manufacturing industry was able to be identified from subcomponents contained in Bill of Material using Machine Learning. A supervised machine learning approach was utilised in this project with the implementation of the Naive Bayes classifier. To further advance the future of this work, the database used to train the model can be improved. Currently the training data was quite simple and small. A multi level BOM data could be incorporated and tested with the established machine learning pipeline. Also, with a larger training data sample, other machine learning algorithms could be tested and compared with the already existing solution.

Furthermore, it is important to test this solution in the infrastructure of the client as a consulting company. Hence, this solution should be tested on the client’s environment and the effectiveness should be measured. This can be done by aligning the BOM Intelligence solution with the client’s KPIs and observing the performance for a specific duration. In addition, this BOM Intelligence solution can be deployed into an interative UI/UX platform for business stakeholders to easily access and make informed data-driven decisions.

# References

Browniee, J. (2014, September 26). *Discover Feature Engineering, How to Engineer Features and How to Get Good at It*. Machine Learning Mastery. https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/

Dear Systems. (2021, September 30). *The Importance of Bill of Materials in Manufacturing*. https://dearsystems.com/the-importance-of-bill-of-materials-in-manufacturing/

Grant, M. (2022, March 13). *Bill of Materials (BOM)*. Investopedia. https://www.investopedia.com/terms/b/bill-of-materials.asp

IHS Markit. (2016). *BOM Intelligence*. https://cdn.ihs.com/www/pdf/BOM-Intelligence-2017.pdf

OpenBOM (openbom.com). (2018, February 9). *OpenBOM Part Catalogs and Bill of Material example*. https://medium.com/@openbom/openbom-part-catalogs-and-bill-of-material-example-2ff29ad8b601

Zhou, C., & Cao, Q. (2019). Design and implementation of intelligent manufacturing project management system based on bill of material. *Cluster Computing*, *22*, 8647–8655. https://doi.org/10.1007/s10586-018-1934-4