The School of Mathematics

THE UNIVERSITY
of EDINBURGH

Scrapp Recycling: Predicting Packaging Parts and Materials

by

Tutku Cinkilic

Dissertation Presented for the Degree of
MSc in Statistics with Data Science

August 2022

Supervised by
Dr. Gordon Ross

Executive Summary

- **Research question:** This study aims to build predictive models for packaging details of products to help Scrapp app verify the unreliable data they have.
- **Data:** Data used in the study is private and provided by the company.
- **Methods:** Multi-label classification methods were performed to model packaging parts and materials. Binary relevance and classification chains methods combined with 3 different algorithms were compared for packaging parts whereas 3 classifiers were suggested for materials according to the specifications of two of the response variables.
- **Results:** Binary relevance model with random forest algorithm and decision tree classifier performed the best for packaging parts and materials respectively. Although the performances of the models were optimistic, they are not enough to verify unreliable data single-handedly.

University of Edinburgh – Own Work Declaration

This sheet must be filled in, signed, and dated - your work will not be marked unless this is done.

Name: Tutku Cinkilic
Matriculation Number: s2259604
Title of work: Scrapp Recycling: Predicting Packaging Parts and Materials

I confirm that all this work is my own except where indicated and that I have:

- Clearly referenced/listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data, etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Not sought or used the help of any external professional academic agencies for the work
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians,and external sources)
- Complied with any other plagiarism criteria specified in the Course handbook

I understand that any false claim for this work will be penalised in accordance with the University regulations (`https://teaching.maths.ed.ac.uk/main/msc-students/msc-programmes/statistics/data-science/assessment/academic-misconduct`).

Signature: Tutku Cinkilic

Date: 19 August 2022

Contents

1 Introduction

One of the biggest reasons why people do not recycle is simply because they do not know how. Scrapp app is a mobile app targeting this problem by offering its users recycling guidelines when they scan the products they want to recycle based on their location and local recycling rules. In order to do that the most reliable packaging information of products comes directly from companies. Another resource is the app's users which is easier to collect but less convenient. Because the user input needs to be verified by the company specialist before being added to their database. This is a manual and time-consuming process that the company would like to automate. In order to check the accuracy of the submitted data, they would like to utilize their on-hand verified data. The on-hand data has several aspects that can be investigated and be useful for the app. This paper will focus on modelling and predicting the packaging parts of products and the materials of these packaging parts.

1.1 Background information about Scrapp App

Before diving into data, we think that explaining how the app works briefly will provide a better understanding of the paper overall. The app is rather simple to use. The user either can directly reach the recycling guide of the product if the scanned product exists in the database or enter the details of the product manually to add the product into the app's system if it is missing. There is a structured flow while filling a product's specifications. The user first types the product's brand, name, and size, later she chooses its category, type, and packaging container from a given list of options. Then based on these options, the suggested packaging parts appear so that the user can also pick a material for each packaging part along with an option to add or remove parts. Finally, the app provides a recycling guide based on the user's inputs.

2 Exploratory Data Analysis

The main data provided by the company and also the primary focus of this study is a table showing the specifications of various commercial products. The table has 9 possible independent variables along with 14 columns of packaging details to be modelled. A row consists of a product's ID, barcode, brand, name, category, type, container, size (amount and unit), and packaging parts in pairs of its type and material. As ID and barcode columns are almost unique for each product, and brand and name columns are textual data, they were excluded from further analyses. From the leftover columns, only size columns is a continuous data. After a quick check, it is seen that the variance of the column (550057.5) is too large compared to its mean (456.4) due to the fact that there are 15 different measurement units. Since this column by itself cannot contribute to any model because of inconsistency, it is neglected as well. In the end, the following columns remain in the table to be examined:

- **productCategory:** (15 levels) Category of the product (food, drink, etc.)
- **productType** (52 levels) Type of the product (alcoholic, carbonated, etc.)
- **container** (50 levels) Container type of the product (glass bottle, plastic bottle, etc.)
- **sizeUnit** (13 levels) Unit to define product's size (L, ml, kg, etc.)
- **packagingPart1-7**: (65 levels) different packaging parts of the product (lid, bottle, etc.)
- **material1-7:** (35 levels) materials of the packaging parts (paper, glass, metal, etc.)

The initial table had 23437 rows. We were informed by the company that the product type column was not always present therefore, products that were recorded earlier have the value of "legacy" to indicate that. There is a total of 3566 rows like this. As it is a substantial amount compared to the data size and these products do not actually come from a similar product type, we omit these rows not to manipulate the possible models. Another omitting action was needed for 784 missing values in sizeUnit column. In the end, 19087 rows were left with all complete values in terms of independent variables.

To start with, we will look into the variables that need to be modelled. There are a total of 7 product parts and material pairs in the table. However, there are no entries with 7 product parts. The majority of products have less than 3 parts while only 23 and 1 products have 5 and 6 parts respectively. The histogram of the number of parts that products have can be seen in figure-1. It is essentially a count data with a mean of 1.969 and a variance of 0.749.

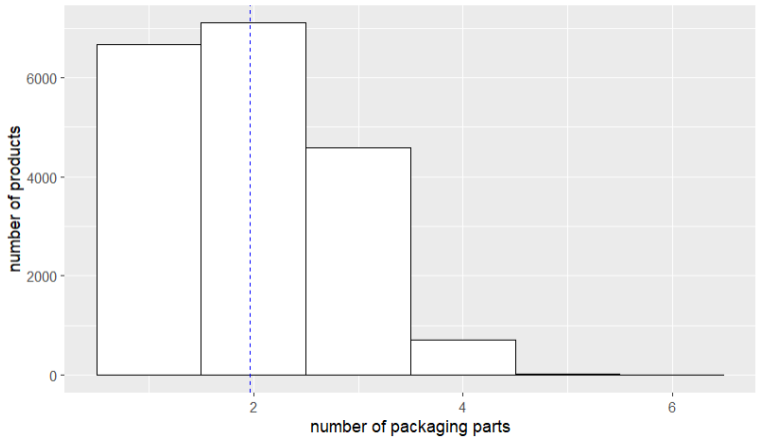

Figure 1: Histogram showing the distribution of the number of parts

The following figure shows the number of missing data in each column. Each product has at least 1 packaging part, therefore packaging part 1 is complete. As expected, the missingness increases

drastically as we move towards packagingPart7. Although modelling packagingPart1 column to predict packing parts might be tempting since all products have to have it, it will not be able to provide all possible parts of a product. Because, while packagingPart1 has 48 levels, there are a total of 65 different packagingPart values are combined in 7 columns. Another interesting aspect of packaging parts is that, except for 37 entries, their values are unique in all 7 columns for a single product. For example, a product has a bottle and a cap but does not have a bottle and a bottle in two different packaging parts columns. Since uniqueness among columns is a desirable property and only 37 entries violate it we delete them from the dataset. This action resulted in a marginal decrease in the mean (1.967) and the variance (0.746) of the number of parts as products with a higher number of parts tend to have the same packaging part multiple times.

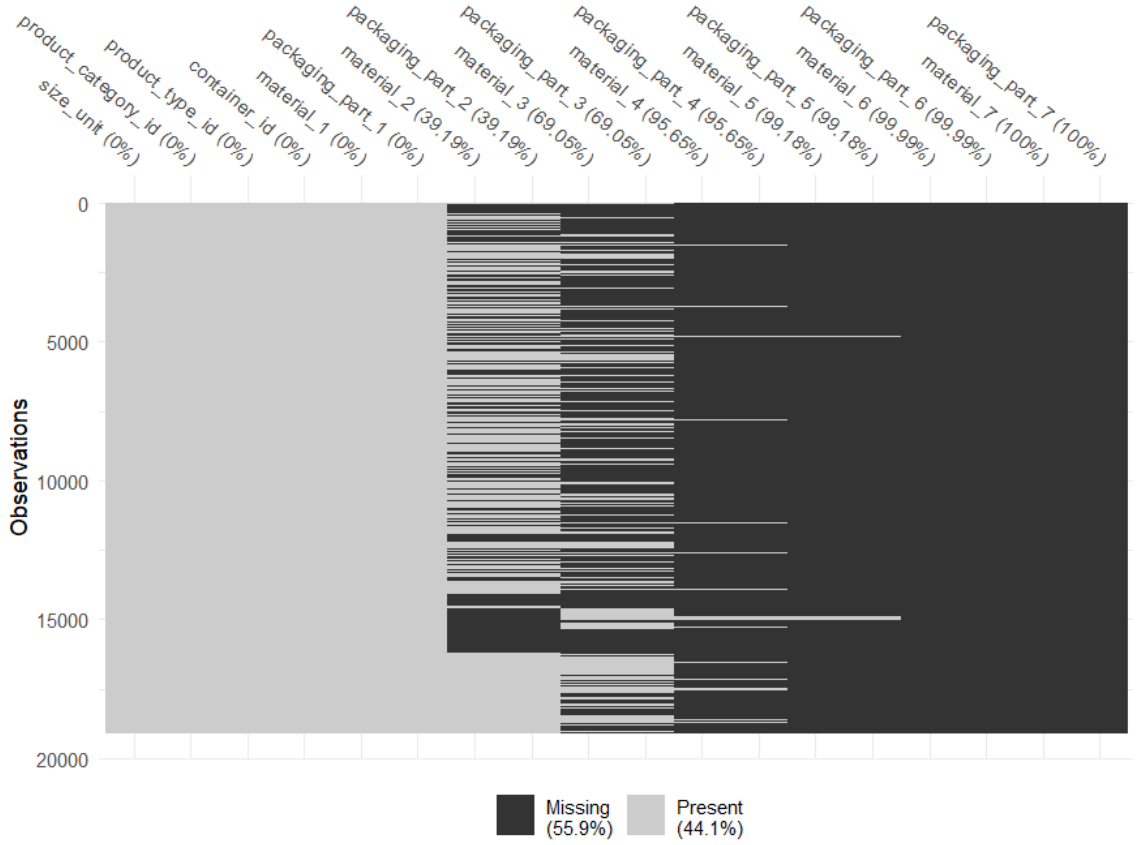

Figure 2: Plot that shows the missing entries in the dataset

Arbitrariness and uniqueness of packagingPart columns for a single product rule out the possibility of modelling only packagingPart1 and expecting the model to perform well. For instance, "box" is a common packaging part that is observed in 5 different packagingPart columns. Products with similar independent variables that should lead to "box" prediction may never be recognised by the model if only packagingPart1 was taken into account and "box" is in any other column but packagingPart1 for those products.

The material columns are different from packagingPart columns in terms of uniqueness and arbitrariness. Multiple parts of a single product may be made from the same material such as plastic. In fact, packagingPart columns become independent variables that can be used while modelling material columns. Similar to packagingPart columns, all levels of materials are not present in material1 column. Only 31 out of 35 materials can be modelled if only material1 column was taken into account.

Moving on to the independent variables, there are a total of 4 which are all categorical. productType and container have 50 and 52 levels respectively. If we plot the histogram of product counts for the different levels of these variables, we see that both have unbalanced product counts at different levels. While some of the levels have thousands of products, some have as little as 2. For these two, we aggregate the levels that have less than 0.1% of the total product count. Dashed lines in figure3and

4 show the cuts for each variable. Since sizeUnit and productCategory already have a small number of levels, we did not perform any merging of less frequent levels.

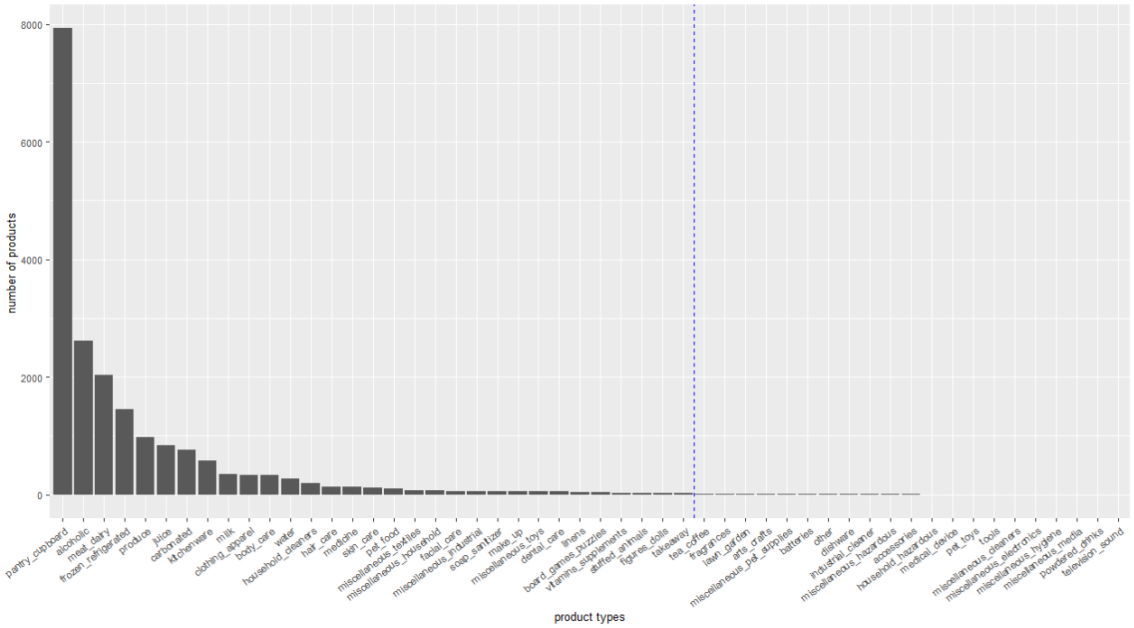

Figure 3: Histogram of product type

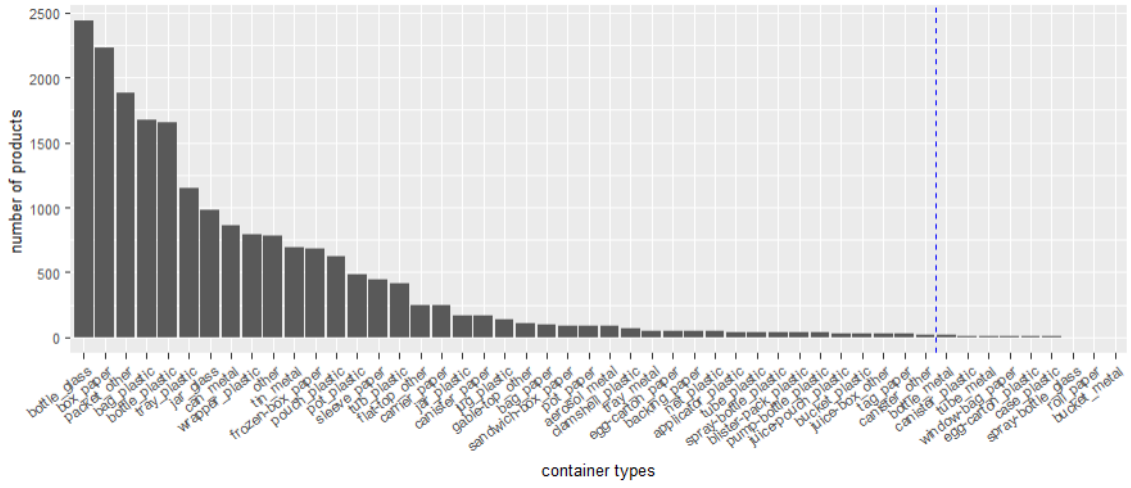

Figure 4: Histogram of container

3 Methods

In section 2, we observed different characteristics of the two variables that need to be modelled. While packagingPart columns were unique and arbitrary for a product, material columns were dependent on packagingPart and were not necessarily unique. These findings lead us to investigate different modelling approaches. Since the material is dependent on packagingPart, the accuracy of the predictions of both heavily relies on the performance of the packagingPart model. Consequently, the modelling of packagingPart has a higher priority in this study. Moreover, we would like to investigate the number of parts per product as it is also an important aspect of the data and closely related to predicting packagingPart and material columns. Hence, this section has three main parts where the modelling of packagingPart, material, and the number of parts are explained.

3.1 Modelling Packaging Parts

Since a product cannot have multiples of the same packaging part, and the order of the packaging columns in the table is arbitrary, we can model all columns at the same time with multi-label classification methods. Multi-label classification is classifying a single entry into a vector of outputs instead of just one. This vector is a fixed length according to the number of different output labels in the dataset [8]. In our case, this means a product will be classified into a vector with 65 binary elements indicating if it includes that packaging part or not.

Various methods are widely accepted in the literature for multi-label classification. These methods are mainly divided into two categories, problem transformation, and algorithm adaptation. While the former transforms the dataset so that single-label classifiers can be used, the latter adapts single-label classifiers to handle multi-label data directly[11]. In this study two compared packagingPart models fall under the problem transformation methods.

3.1.1 Models

The binary relevance model is one of the most common and intuitive solutions to the multi-label classification problem. The idea behind this method is to divide the multi-label task into several independent binary classification tasks. One obvious weakness of this approach is to ignore the dependence between different labels [17]. For example, tub-lid, bottle-cap, and box-wrap are a few of many packaging pairs that are commonly observed together. However, this method ignores these co-appearances. One way to overcome this handicap is to allow a chain structure between the separate binary models.

Classifier chains (CC) are known as a correlation-enabling extension of BR model[17]. Again in this method, the same number of binary models are fitted to data as in BR, however, this time predictions are made based on the preceding models' outputs. In other words, with every binary assignment, feature space grows by one variable and the next binary model is fitted to this updated larger dataset.

Data Preparation: To apply these 2 models, we have altered the dataset so that instead of 7 packagingPart columns, 65 binary columns are added, where each stands for a different packaging part level. The elements in the columns take a value of 1 if the product has the specific packaging part, and 0 otherwise. As mentioned in 1.1 , in order to leverage all information given by the users, we include all categorical variables explored in section 2. At last, the dataset that is used to train models has 69 columns in total, 4 for productCategory, productType, container ,and sizeUnit; 65 for the binary outputs.

3.1.2 Algorithms

For the binary classification of each label column which is necessary for both methods, three commonly used single-label classification algorithms were used. These are explained in the following paragraphs briefly as the main goal of this study is accurate predictions rather than the theoretical aspect of the used algorithms. Please refer to references for further details.

Decision tree (DT) classifiers have tree-like structures which separate a dataset into smaller parts. Using features of entries, data is classified starting from the root node until it reaches a leaf node which represents a decision. They are widely used due to their ability to handle both categorical and continuous data and their interpretability. The specific decision tree algorithm used in this paper is C5.0 [10] which is an enhancement of the previous version C4.5[14] that is based on the first proposed version ID3[12].

Random forest (RF) is an ensemble of multiple decision trees. An entry is classified by all decision trees in the model, and the final decision is the most common output among the decisions made by all trees. Contrary to DTs, RF does not provide an interpretable tree-like structure. However, Its efficiency and flexibility make it one of the most used classification models[2]. The random forest algorithm we implemented on R is still based on the original code of Breimen [4].

Naive-Bayes (NB) classifier is a probabilistic classifier that is based on the assumption that given the response values, the probability of observing the combination of explanatory variables is simply

the product of their individual probabilities. It completely ignores the correlation between the model variables and assumes independence[9]. This property makes the model undesirable for many real-life situations. However, since we are more into predictive performances of alternative models rather than a perfect fit and our data is categorical, we'll still use the algorithm and assess its performance.

In the end, 65 separate binary classification task was performed to make predictions as a base model(BR) and also a classifier chain (CC) model was fitted where classifiers take the output of the previous classification task as an input. The methods BR and CC are implemented on R with "utiml" package using the algorithms from "C50", "e1071", "randomForest" packages.

3.2 Modelling Materials

Material columns differ from packagingPart columns in terms of uniqueness. Therefore, we cannot use BR or CC as they are used for binary classification of labels. However, it is still a multi-label case in a way that each product is allowed to be associated with more than one label. Similar to BR and CC, there is another problem transformation method called copy transformation [5] which turns a multi-label classification task into a single-label classification task. Unlike BR and CC, this method allows multi-class classification.

Copy transformation duplicates the entry as many times as the number of labels it has. Let Figure-5 (a) symbolize our dataset with four entries. λ_i demonstrate packagingPart-material pairs. After applying copy transformation, the dataset becomes Figure-5 (b). In our case duplicated rows are not totally the same because of each unique packaging part that defines the material. So, the packaging parts which were the dependent variable to be modelled in the previous section are now independent variables. However, the same approach still can be adapted to transform the problem. After the transformation, the variable that needs to be modelled is one large material column with all different material classes. The same independent variables are used as packagingPart models.

Example	Attributes	Label set
1	$\mathbf{x_1}$	$\{\lambda_1, \lambda_4\}$
2	$\mathbf{x_2}$	$\{\lambda_3, \lambda_4\}$
3	$\mathbf{x_3}$	$\{\lambda_1\}$
4	$\mathbf{x_4}$	$\{\lambda_2, \lambda_3, \lambda_4\}$

(a) Example dataset

Ex.	Label
1a	λ_1
1b	λ_4
2a	λ_3
2b	λ_4
3	λ_1
4a	λ_2
4b	λ_3
4c	λ_4

(b) Data set after copy transformation

Figure 5: Two tables showing copy transformation as seen on [1]

After the transformation of the dataset, three different single-label multi-class classifiers are fitted to the data. These are Multi-Nomial logistic regression (MN-LR), Naive-Bayes classifier (NB) and C4.5 decision tree (C4.5). NB and DT algorithms are as explained in section 3.1, also the same packages were used to implement them.

MN-LR is simply an extension of logistic regression which is a widely used generalized linear model for binary response variables. The response in MN-LR has more than 2 categorical classes which are unordered[15]. MN-LR is preferable over other methods for its ability to fit data better and produce more proper insights even though some common assumptions in modelling such as independence of variables or normality of response are violated [3].

The models were implemented on R. For MN-LR model, "multinom" function from package "nnet" was used.

3.3 Modelling Number of Parts

As seen in section 2, the number of parts is a count data with a mean of 1.967 and a variance of 0.746. The straightforward modelling option for count data is Poisson regression. Poisson regression (PR) is a generalized linear model where the response variable is Poisson distributed and the logarithm of

the expected value of the response variable is a linear function of independent variables. However, Poisson regression assumes equidispersion which means the mean equals to the variance. In our case, under-dispersion is present where the variance is smaller than the mean[7]. Although Poisson regression is an inappropriate choice for our response, we just want to use it as a baseline model to compare the number of parts predicted by binary packagingPart models. We would like to see that the packagingPart models are at least performing as well as a model which does not even satisfy all of its assumptions. The regression formula is as in 3.1 where λ_i is the expected value of the response, X is the covariate matrix and β is the vector of coefficients. Again, all four categorical variables are included in the model.

$$log(\lambda) = \beta X \tag{3.1}$$

R is used to fit PR by "glm" function.

4 Results

Results of the models will be examined in three separate sections following the structure of the methods section.

4.1 Results of packagingPart models

Arbitrarily placed, packagingPart columns with a unique set of packaging part values were modelled using multi-label classification techniques. 6 alternatives were examined with a combination of 2 methods and 3 algorithms. Data were split into training and testing sets to assess the models' predictive performances. The outcomes of test sets were used to make comparisons. All results can be found in figure-6.

Assessing the performance of multi-label classification is a little trickier than single-label classification as the prediction can be partially true instead of a certain yes or no. We can calculate measures based on either per-instance or per-label [13], then take their average as overall performance. Here, we chose the per-instance approach as we are more concerned about per-product metrics rather than individual labels.

Let true positive (TP) be the number of truly predicted 1s, false positive (FP) be the number of wrongly predicted 1s, true negative (TN) be the number of truly predicted 0s, true negative (FN) be the number of wrongly predicted 0s. The metrics shown in 6 are calculated as follows [16].

- **accuracy:** $\frac{TP+TN}{TP+TN+FP+FN}$
- **precision:** $\frac{TP}{TP+FP}$
- **recall:** $\frac{TP}{TP+FN}$
- **F1-score:** $2 * \frac{Precision*Recall}{Precision+Recall}$
- **Hamming-loss:** $\frac{FN+FP}{TP+TN+FP+FN} = 1 - accuracy$
- **subset-accuracy:** Also known as exact match ratio is the most strict measure out of them all. It is the ratio of exact true predictions to all.

Coloring in 6 shows the top three performers for each metric. BR-RF is the clear winner by scoring the best in all. It is a bit surprising since we were expecting CC method to perform better as it takes dependency between labels into account. The second best is again RF algorithm but with CC method this time. We can comment that RF algorithm performed much better in binary classification as it beat the combination of the winner method and other algorithms with a less performing method. However, one can see that results do not differ significantly between the two methods for the same algorithm. In terms of accuracy NB scores much worse than the other two. As stated in section 3.1, the NB model strictly assumes the independence of features. As all data on hand are categorical we

could not clearly assess that. Yet this result might be indicating some dependency. DT also performs mostly just under RF. It was rather an expected outcome as RF is a method using multiple DTs to decide whereas DT uses single. As it also has promising scores, one can investigate its structure to better understand the data. In terms of sub-set accuracy, NB scored extremely poorly around 30%, and even the best score is only 61%.

Method	Algorithm	Set	accuracy	F1	hamming-loss	precision	recall	subset-accuracy
BR	DT	Train	0.801	0.8587	0.0089	0.9223	0.8341	0.6042
		Test	0.7922	0.8523	0.0095	0.9181	0.827	0.5896
	RF	Train	0.8187	0.8722	0.0081	0.9285	0.851	0.6368
		Test	0.8068	0.8634	0.0089	0.9204	0.8426	0.6149
	NB	Train	0.6876	0.7725	0.0178	0.7826	0.8305	0.3979
		Test	0.6896	0.774	0.0175	0.7866	0.8281	0.4001
CC	DT	Train	0.8023	0.8585	0.0091	0.9093	0.8431	0.6177
		Test	0.7937	0.8525	0.0097	0.9039	0.8365	0.6023
	RF	Train	0.8013	0.8576	0.009	0.9186	0.8341	0.6117
		Test	0.7951	0.8533	0.0094	0.9144	0.8296	0.6004
	NB	Train	0.5945	0.6985	0.0256	0.6375	0.8394	0.2678
		Test	0.6004	0.7028	0.0251	0.645	0.8373	0.2794

Figure 6: Comparison table of packagingPart models. The top 3 best performing models in each metric is marked from dark blue to light blue, dark blue showing the best result.

4.2 Results of material models

To implement material models, we applied copy transformation to the dataset and transformed predicting multiple materials columns into a multi-class single label problem. Therefore, to assess the model performances, we'll use some multi-class classification metrics as seen in [16]. The calculation of these metrics only differs from the previous section in terms of how they are averaged over. As only a single label is predicted, we calculate the metrics for each class and then take their average for comparison.

7 shows the results of material models. Again, there is an obvious winner. DT is better than the other two in all metrics except for recall. Interestingly, NB tends to score higher recall in both packagingPart and material models. MN-LR has closer scores to DT yet it is outperformed in all metrics by another algorithm.

Algorithm	Set	accuracy	precision	recall	F1
MN-LR	Train	0.7641	0.7928	0.5718	0.6653
	Test	0.7584	0.6366	0.4893	0.5928
DT	Train	0.7891	0.8344	0.5564	0.6706
	Test	0.7782	0.7202	0.4707	0.6324
NB	Train	0.6615	0.6163	0.6209	0.5743
	Test	0.66	0.5575	0.5157	0.5414

Figure 7: Comparison table of materials models. The top 3 best performing model in each metric is marked from dark blue to light blue, dark blue showing the best result.

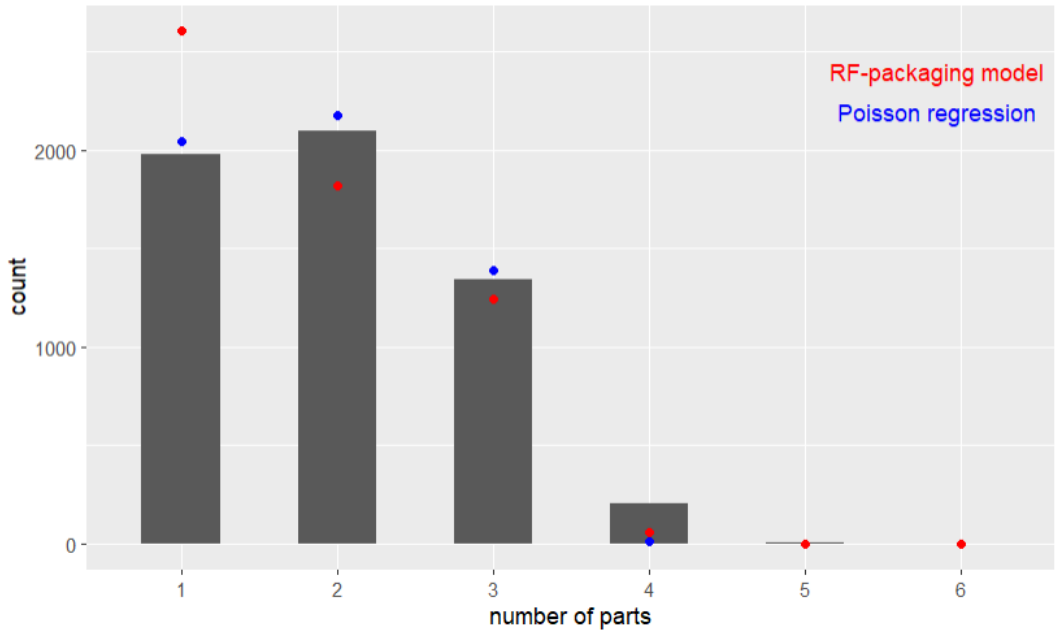

Figure 8: A histogram showing the true counts of the number of parts in the dataset. Predicted counts of BR-RF and Poisson regression are marked with dots of different colors.

4.3 Results of the number of parts model

We fitted a Poisson regression model to the number of parts data even though we detected that the data did not have equal mean and variance. In this situation, we expect packagingPart models to perform better than the regression model. As each classified label in packagingPart models stands for a packaging part, these binary multi-label models indirectly predict the number of parts per product as well. We can sum 1s across rows and obtain the number of parts that is assigned to each product. For comparison we choose the best performing packaging part model, BR-RF, assuming that it will also be the best performing in terms of the number of true assignments.

From 8, we can see that poisson model follows the pattern of the count much better than BR-RF, however it tends to underestimate the number of parts. As one can see, the highest prediction is 4, and the number of products assigned to 4 is very little compared to the real number. On the other hand, RF is completely off in terms of the number of parts. It cannot grasp the bell shape of count distribution and clearly assigns most of the products only one packaging part label.

As the predictive performance of the model is the main concern, estimated regression coefficients that indicate the relationship between independent variables and the number of parts will not be discussed in this report.

5 Conclusion

The main goal of the project was to build predictive models for packaging parts and materials of products to help the company verify the data coming from their users. For this purpose, packagingPart and material columns were transformed by multi-label problem transformation methods. Total of 6 and 3 alternatives were proposed for packagingPart and materials respectively. While BR-DF outperformed all packagingPart models, DT was the best among material models. Although both models are promising with high accuracy scores, they are not solely enough to verify user inputs. The subset-accuracy score of BR-RF is 61%, and since the material model takes packagingPart as an input, prediction of both at the same time requires two-step predictions, first the packaging parts and then the materials taking packaging part predictions as input. Therefore, predictions can only be as accurate as the packaging part model.

5.1 Discussions

The suggested models can also be used as a recommendation feature in the app as users fill out the specifications of products. It is safe to assume that if a user sees the 10 most possible packaging parts on top of packaging parts options for the product, she has already specified its type, category and container, she will be less likely to forget about a part such as the label of the container which is usually overlooked.

5.2 Limitations

To fit multilabel models to the packaging parts 37 entries with duplicated packaging parts were removed. Therefore all suggested models assume the uniqueness of packaging parts per product and will not be able to handle the duplication of parts situation.

5.3 Further research

It was an exciting project in terms of the variety of areas that could be worked on. This paper focused on predicting packaging parts but there are many more topics that can be investigated. Some of these ideas are listed below.

- We have implemented binary relevance and classifier chain methods with the default settings of the software packages. Different settings can be tried for optimal performance. For example, while implementing CC, the order of the chain was random as default in the package. The optimal order might be investigated and better results can be achieved Due to the time constraint of this project, other single-label classification algorithms were not examined.
- Feature selection strategies can also be researched which might improve overall model performances. Preprocessing of the data was limited to aggregating less frequent levels of categorical variables. However, still container variable had more than 40 levels.
- The textual aspect of the data was excluded from the scope of this project. In fact all the independent variables that are used to make predictions and also omitted columns such as brand and product name are textual data. These can be merged into larger strings and text classification methods can be investigated in further studies.
- Another aspect of data that might be an interesting topic to examine might be the possible missingness of the columns that are assumed to be complete in this paper. We assumed product type, category and container are all given and complete. But in the case of users filling up this information they might also be unreliable and missing.

References

[1] *Data Mining and Knowledge Discovery Handbook*. Springer US, 2010.

[2] J. Ali, R. Khan, N. Ahmad, and I. Maqsood. Random Forests and Decision Trees. Technical report, 2012.

[3] A. Bayaga. MULTINOMIAL LOGISTIC REGRESSION: USAGE AND APPLICATION IN RISK ANALYSIS. Technical report.

[4] L. Breiman. Random Forests. Technical report, 2001.

[5] R. Buch and D. Ganda. A Survey on Multi Label Classification AN AUGMENTATION OF TCP FOR COMPETENCY ENLARGEMENT IN MANET View project Cyber Security View project Recent Trends in Programming Languages A Survey on Multi Label Classification. pages 19–23, 2018.

[6] M. Grandini, E. Bagli, and G. Visani. Metrics for Multi-Class Classification: an Overview. 8 2020.

[7] T. Harris, Z. Yang, and J. W. Hardin. Modeling underdispersed count data with generalized Poisson regression. Technical Report 4, 2012.

[8] F. Herrera, F. Charte, A. J. Rivera, and M. J. Del Jesus. Multilabel Classification Problem Analysis, Metrics and Techniques. Technical report.

[9] M. J. Islam, Q. M. Jonathan Wu, M. Ahmadi, and M. A. Sid-Ahmed. Investigating the performance of naïve- bayes classifiers and K- nearest neighbor classifiers, 2010.

[10] A. Mona Nasr, E. Shaaban, R. Pandya, J. Pandya, and K. Infotech Amreli. C5.0 Algorithm to Improved Decision Tree with Feature Selection and Reduced Error Pruning. Technical Report 16, 2015.

[11] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. Merschmann. Categorizing feature selection methods for multi-label classification. *Artificial Intelligence Review*, 49(1):57–78, 1 2018.

[12] J. R. Quinlan. Induction of Decision Trees. Technical report, 1986.

[13] A. Rivolli and A. C. P. L. F. De Carvalho. The utiml Package: Multi-label Classification in R. Technical report.

[14] S. L. Salzberg. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning*, 16(3), 1994.

[15] P. J. Smith. Analysis of Failure and Survival Data. Technical report.

[16] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4):427–437, 7 2009.

[17] M. L. Zhang, Y. K. Li, X. Y. Liu, and X. Geng. Binary relevance for multi-label learning: an overview, 4 2018.

[18] M. L. Zhang and Z. H. Zhou. A review on multi-label learning algorithms, 2014.

Code

Codes used to provide the result presented in this paper can be found in the following link. Python is used for the initial analysis and models are fitted in R.

- Code link