Deakin SIT Research Project

Classification of Grocery Products with Random Forest and Rule-Based Method

Submitted as Research Report / Master Dissertation in SIT723

October 10th

T2-2021

Kriutchenko Alina

STUDENT ID

220129254

COURSE - Master of Data Science (S777)

Supervised by: Prof. Jean-Guy Schneider, Ms. Josephine Marshall,
A/Prof. Kathryn Backholer

Abstract

Nowadays people want to consume healthy grocery products, but they find it difficult to select them without meticulously inspecting their nutrition information. The shelves are full of colorful packages that draw people's attention. In this article, we try to classify grocery items using ML to create a solution that could improve the shopping experience for those who take food quality seriously (Willis, 2019).

This is a relevant problem because with so many options buyers often chose more visually appealing, cheaper, and less healthy options. This method would improve the food supply process. This will allow health organizations to monitor the nutritional value of the food and beverages supplied by supermarkets, as well as their availability, price, and advertising. To do so, the products must be automatically categorized according to their healthiness. This is now done manually, which is a time-consuming process. Nutritionists need to apply the created method to automatically classify all new things based in the data. There are various advantages of using automatic categorization. This will improve overall population health.

The approach to this problem is a comparison of two classification methods: Rules-Based Classifier for text data and Random Forest classifier for numerical data. The Rule-based method applied to text data such as product name and brand. We tested it with a set of rules that will be expanded later. In this case, the accuracy result is not as essential as an effectiveness of rules. The Random Forest Classifier applied to the numerical nutrition data such as 'energy', 'protein' and 'fat'. The accuracy metrics shows a high result over 90% for prevalent classes, while other classes had significantly lower accuracy mainly caused by lack of data. It is possible to classify the healthiness of grocery items by using ML techniques. However, there are many different aspects that should be taken into consideration: unique, unknown product and class unbalanced.

Overall, the Rules-Based Classifier with text data may promise a higher accuracy for a selected supermarket chain. This research has a limited number of rules, but with more rules in the future, it has a potential to achieve an almost 100% accuracy. The Random Forest Classifier would be expected to work better on a randomly selected product list of foods and beverages, and there is no language barrier for the numerical data.

Keywords: Classification ● Rule-Based Classifier ● Random Forest ● Health Ranking

Content

1 Introduction

This research aims to explore the possibility to classify grocery products by healthiness level. We focus our attention on Rule-Based and Random Forest (RF) classifiers. We explore two possibilities for classification: text data and numerical data. Both methods have different approaches and limitations. The target variable is 'ADG', that represents healthiness.

1.1 Background:

Most grocery stores do not provide health ranking of the product or provide it only for items of its own production. Therefore, for most people it is complicated to select healthy food without any domain knowledge of its benefits. Without strict classification it is very easy for supermarkets to supply mostly unhealthy products.

One in four Australian children is overweight, so it is important for regulatory institutions to be involved in the assessment of food healthiness (The University of Sydney, 2017). Choosing foods that are higher in positive nutrients and lower in risk nutrients that are linked to obesity and diet-related chronic diseases. (Healthstarrating.gov.au, 2019)

We decided to find a way to automatically assign a healthiness classification according to the Australian Dietary Guidelines (ADG) to grocery products. The list of grocery items from Coles and Woolworth website collected by Web scraping. The automatic classifier would allow for researchers to investigate the healthiness of food supplied for various supermarket chains.

1.2 Research problem:

The main problem is to find if there is a possibility to automatically classify product based on healthiness. The manual assigning of food labels requires a lot of time from nutritionists. With so much available data, there is also a risk to misclassify some items.

The machine learning methods could be effective to solve this problem. This should be approached by using information about the product collected from the supermarket's website.

1.3 Objective(s):

The main objective is to investigate the possibility of predicting product healthiness by product name, brand, category, or nutrition data. The goal is to investigate the possibility of classifying product healthiness based on a different set of variables with two different methods.

This solution would improve the food supply strategies and regulations rather than displaying the healthiness for consumers. This way heath institutions can monitor the healthiness of the food and beverages sold by supermarkets, including their availability, price, and promotion. To do this, the products need to be classified by healthiness. Currently this is completed manually, but this is a time-consuming process.

The goal of nutritionists is to apply the created approach to automatically classify all new items based on their data. There are various advantages of using automatic categorization instead of manual classification. The most important benefit is that it saves time. To manually classify all these products would take a significant amount of time that could be spent on more important duties. Furthermore, the manual classification has various hazards, such as misguiding the product's healthiness due to its name.

1.4 Solution

The random forest classifier uses the nutritional data such as Energy, Protein, Fat, Saturated Fat, Carbohydrate, Sugar and Sodium. The algorithm finds connection between variables, trying to identify the category.

The rule-based classifier follows the suggestions that if a product name, brand, or category contains or does not contain certain words or combinations of character, then the variable is considered a part of a suggested class.

The random forest classifier is capable to be accurate on some more obvious classes (healthy and unhealthy) but have difficulties classifying classes that were assigned taking into consideration semantic meaning of the product in addition to nutritional data.

The rule-based classifier does not have a reference of any class and is capable to classify most products with the meaningful set of rules.

1.5 What follows

The literature review (section 3) includes three parts. The regular expressions that we use to clean the data: it includes splitting the digits and character of the nutrition column. The rules-based classifier part covers an overview of the strategy behind the method. The literature review also includes the selection of the classifier for numerical data. It explains why the Random Forest was selected for this problem.

The section 4 dedicated to research design and methodology. First, it covers the data structure. The section 4.2 section includes the description of the data cleaning process and the regular expression application to the data. In section 4.2 we discussed the reasoning behind the train and test split for the random forest classifier. Section 4.4 describes the process of parameter selection for the RF classifier. In section 4.5 we covered the techniques used to assess the classifier quality such as classification report and confusion matrix. The last part is about predictions.

Section 5 explains the process of the experiments performed. This included specifics of rules design and how the classifier uses it. The number of rules included in this test is under 100 rules.

In Section 6 we present the results of our classification. It includes comparisons by class, overall accuracy, and potential conclusions from it.

Section 7 discusses potential limitations of developed solutions. It touches on the data characteristics, and restrictions of both classifiers based on aspects like store location and time.

Finally, in section 8 we discuss conclusions, the reasons behind it and where we can follow from there.

2 Properties

The original data has 11253 observations and 27 variables. **The list of variables** is: 'Source', 'Brand', 'Product name', 'Category', 'Pack size', 'Serving size', 'Servings per pack', 'Energy per 100g (or 100ml)', 'Protein per 100g (or 100ml)', 'Total fat per 100g (or 100ml)', 'Saturated fat per 100g (or 100ml)', 'Carbohydrate per 100g (or 100ml)', 'Sugars per 100g (or 100ml)', 'Sodium per 100g (or 100ml)', 'UID', 'Product code', 'Nutrition info from match', 'Match source', 'Match UID', 'Dietary code', 'Food category', 'ADG (1=ffg,2=disc, 3=other, 7=special foods, 8=unable to categorise, 9=na)', 'Beverage category (1=SSB, 2=ASB, 3=FFG SSB, 4=plain water, milk, 5= coffee/tea, 6=bev mixes)', 'NIP updated (1=added info when none previously, 2=updated existing value, 3=other info updated (red text))', 'date NIP updated', 'NIP source', 'Notes'.

The grocery data differ from common reviews classification problem. Therefore, instead of sentiment analysis, we decided to develop a rule-based classifier with a unique set of rules. (Volodymyr Bilyk, 2019) The numerical classifier (Random Forest) selected based on its effectiveness for the datasets with highly unbalanced classes.

For the **Rules-Based classifier** we select: 'Source', 'Brand', 'Product name' and ‘Category'. All of them are categorical nominal variables and have no missing data. For the **Random Forest Classifier**, we use nutrition data: 'Energy per 100g (or 100ml)', 'Protein per 100g (or 100ml)', 'Total fat per 100g (or 100ml)', 'Saturated fat per 100g (or 100ml)', 'Carbohydrate per 100g (or 100ml)', 'Sugars per 100g (or 100ml)', 'Sodium per 100g (or 100ml)'. All of them are continuous numerical variables. The nutrition columns have over 40% of missing observations. For this experiment we had only 5860 observations and this creates some limitations.

The target variable is **ADG** (1=ffg,2=disc, 3=other, 7=special foods, 8=unable to categorize, 9=na). The classes are defined by Australian Dietary Guidelines classification: ffg (healthy), disc (less healthy foods), other foods (such as tea, coffee, spices), special foods (such as alcohol, vitamins, baby), unable to categories (such as meal kits, packages), na (non-food).

3 Literature Review

3.1 Data Preprocessing: Regular Expressions

The article describes the patterns used in regular expressions (thepythonguru.com, 2020). For our problem we created 2 loops with patterns for the selection of characters and digits (Real Python, 2019).

Regular expressions (called REs or regexes) are essentially a tiny, highly specialized programming language embedded inside Python and made available through the **re** module. The regular expression is a pattern that matches strings or pieces of strings. The set of strings that they can match goes way beyond what regular expressions from language theory can describe (Kuchling, 2019).

For instance, the regex "\d" is select values {0,1,2,3,4,5,6,7,8,9} from the column value. They are supported by most of the programming languages like python, perl, R, Java and many others (Ray, 2015).

3.2 Rules-Based Classifier

Rule-based is just another type of classifiers which makes the classification decisions depending by using various "if..else" rules. These rules are easily interpretable and thus these classifiers are generally used to generate descriptive models. The condition used with "if" is called the antecedent and the predicted class of each rule is called the consequent. (thepythonguru.com, 2020)

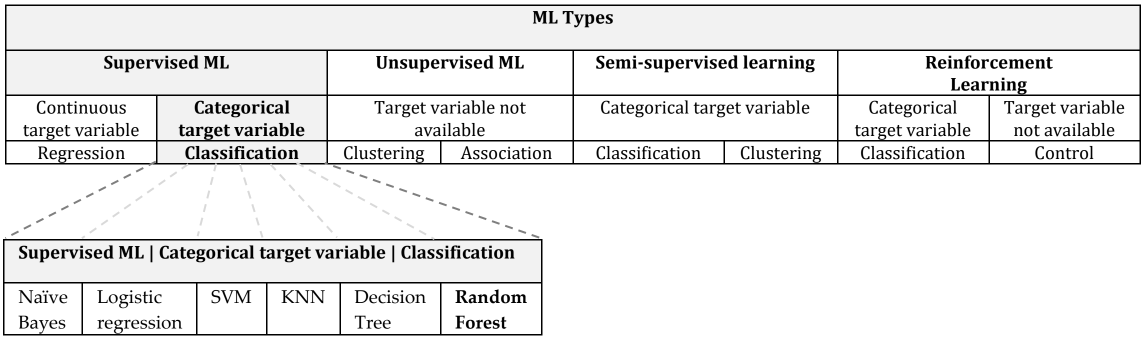
Properties of rule-based classifiers:

- Coverage: The percentage of records which satisfy the antecedent conditions of a particular rule.
- The rules generated by the rule-based classifiers are generally not mutually exclusive, i.e., many rules can cover the same record.
- The rules generated by the rule-based classifiers may not be exhaustive, i.e., there may be some records which are not covered by any of the rules.
- The decision boundaries created by them are linear, but these can be much more complex than the decision tree because many rules are triggered for the same record. (Virmani, 2020)

The obvious question that arises after learning that the rules are not mutually exclusive is how the class would be determined if different rules with different consequences applicable to the record. (Bizer, 2020)

3.3 Random Forest

The article provides insights on ML algorithm selection (Gavrilova, 2020). In our problem with having a target variable, therefore we consider supervised algorithms (Brownlee, 2020). The target variable is "adf' that is a categorical variable with 7 classes. In the table 1 we can see the most popular algorithms for data with categorical target variable (Wolff, 2020).



ML Types							
Supervised ML		**Unsupervised ML**		**Semi-supervised learning**		**Reinforcement Learning**	
Continuous target variable	**Categorical target variable**	Target variable not available		Categorical target variable		Categorical target variable	Target variable not available
Regression	**Classification**	Clustering	Association	Classification	Clustering	Classification	Control

Supervised ML | Categorical target variable | Classification					
Naïve Bayes	Logistic regression	SVM	KNN	Decision Tree	**Random Forest**

Table 1. ML Types and Algorithm Selection

The research on Random Forest provides evidence of being more successful on dealing with the lack of data or unbalanced classes.

- The **Naïve Bayes** is more suitable for Text Classification.
- **SVM** performs optimally in cases where there is a distinct margin of separation among classes. By default, they are not effective at imbalanced classification (Brownlee, 2020)
- **KNN**: Imbalanced class size is a problem KNN has been characterized by.
- **Decision Tree**: can create biased learned trees if some classes dominate. Prone to overfitting. (Yadav, 2018)
- **Random Forest**: Solves the issue of overfitting in DT. More effective with unbalanced data. (Chen and Liaw, n.d.)

4 Research Design & Methodology

4.1 Data Overview

The Data we use for Random Forest Classifier is 7 nutrition columns with numerical continuous values. The data we use for Rule-based Classifier is ‘Brand’, ‘Product name’, ‘Category’ columns with categorical nominal values.

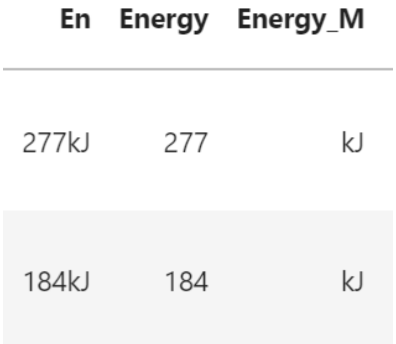
4.2 Data Pre-processing

The data cleaning stage includes several tasks.

One of them is dealing with “less than” or “<” values. Based on the nutritionist’s recommendations is decided to take the value that is 50% less than the original number. For example, “<0.1” converted to “0.05” (Tableau, n.d.).

One of the most important methods is Regex. Regular expressions are used to split the numerical value and the measurement value. (Finxter, 2020). For instance, “re.findall” used to select digital from the value like “13.5g” to receive the continuous numeric value “13.5” (Table 2), (Sanad Zaki Rizvi, 2020).

Original Column ‘En’ is split into 2 columns: with numeric and character values.



En	Energy	Energy_M
277kJ	277	kJ
184kJ	184	kJ

Table 2. Example of Regex Split

The data also include about 80 rows that contain complex and unique values that require manual transformations. For example, creating a rule for this or similar value might lead to receiving incorrect data. There, we manually selected the “4.2g” of protein based on “Taco Kit”.

There is a certain number of missing values in the dataset. To make the pre-processing more convenient, we temporary filled the missing values with ‘111111novalue’ that contains both digital and non-digital characters. We removed them later.

4.3 Train/Test Split:

The dataset is relatively small and contains unbalanced classes. Therefore, the selected ratio of train/test split using a stratified sampling method: 0.4 test set for every class. (Tokuç, 2021)

4.4 Parameter selection

The parameter selection for Random Forest classifier described in section 4.2. The parameters applied were selected with consideration of the dataset that we have. There is a list of these parameters and some of its characteristics that played a role in our choice later.

Criterion: Gini impurity tends to isolate the most frequent class in its own branch of the Tree, while entropy tends to produce slightly more balanced Trees (Goyal, 2021). The selected criterion="entropy", Accuracy for class 7 slightly higher than the accuracy with the default "gini"

min_samples_split: When we increase this parameter, each tree in the forest becomes more constrained as it must consider more samples at each node (Fraj, 2017). The value considered is close to default 2.

Bootstrap: It means that some samples will be used multiple times in a single tree. It is not very useful for some smaller classes that we have since it creates biased data points (Koehrsen, 2018). Therefore, bootstrap=False

N_jobs: It specifies the CPUs used and algorithm's parallel running. The number is increased from default 1 to add more stability.

Random_state: It prevents randomness and fixed highest results.

class_weight: It changes the weights to correct class imbalance. (Albon, 2017)

4.5 Assessment methods

We use accuracy and recall assessing the performance of Random the classifiers. (Koo Ping Shung, 2018)

- **Accuracy** is the most intuitive performance measure, and it is simply a ratio of correctly predicted observation to the total observations.
- **Recall** (Sensitivity) is the ratio of correctly predicted positive observations to all observations in actual class - yes. Recall = TP/TP+FN.
- **Precision** is the ratio of correctly predicted positive observations of the total predicted positive observations.
- **F1 Score** is the weighted average of Precision and Recall.

For the assessment of the Rule-Based Classifier we are going to use overall accuracy, but without analysis by classes because this research has limitations related to a limited set of rules.

The correct and incorrect **prediction** will be manually analyzed. In the Rule-Based classifier, it will include assessment of the correct application of rules.

5 Experiments

5.1 Rules Based Classifier

The Rule-Based Classifier performed on the text data. We created a function that goes through a list of rules that are based on 3 columns: Brand, Product Name, or category. The predicted variable is “ADG” or “Healthiness” Variable. If the product in the data matches the rule condition, it assigned to be a certain class. If it does not match any condition, the product is labeled as a default class 2, which means unhealthy product. For instance, when the Brand contains “Angus Park”, but Product name does not contain “Chocolate” then the product assigned to category 1, which is a healthy product.

The Rule-Based classifier is created inside the **apply_rules** function. The function has a multiple **”if”** statements, that based on the decision tree classifier. (Molnar, n.d.) Every **condition** matches a certain combination of characters in one of 3 columns: “Brand”, “Product name” and “Category”.

apply_rules function: **def apply_rules (source, brand, product_name, category)**

When the row matches a certain condition, one of these categories is being selected:

```
ADG_FFG = 1
ADG_OTHER = 3
ADG_SPECIAL = 7
ADG_UNABLE = 8
ADG_NA = 9
```

Otherwise, it assigns the default category:

```
ADG_DISC = 2
```

Every rule can have one or multiple conditions. Regex applied to search for a pattern. We use **“re.search”** to match a combination of characters in the value or we use **“test and re.match”** to match a value as a whole. Also, we use **“not re.search”** to exclude some matches from the list that selected based on the previous conditions. The function scrolls through rows in our dataset while assigning the class based on matches in rules.

Rule Example: Another **Rule example**: if the Brand contains ‘angas park’, but the product name does not contain ‘chocolate, the Product is ‘healthy’. The first letter is searched in both upper case and in lower case.

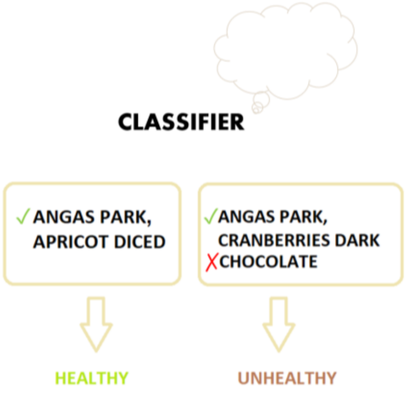

CLASSIFIER
✓ANGAS PARK, APRICOT DICED
✓ANGAS PARK, CRANBERRIES DARK ✗CHOCOLATE
HEALTHY
UNHEALTHY

Figure 1. Classifier illustration

To assess the classifier accuracy, we added additional column ‘Result’. It produces ‘True’ value when the predicted value matches the actual value. Otherwise, it produces ‘False’. The ratio of True/False at the final experimentation stage: 60.92 (TRUE), 39.08 (FALSE).

Rule examples that produced **correct** (‘True”) results:

Example #1

If brand contains 'John west' (upper or lower case), the product is ‘healthy’ (class 1). We have a prediction of ‘1’ that matches the original value. The Result column (Table 3) gets the value ‘True’ (correct prediction).

```
if (re.search ("[Jj]ohn west", brand)):
    adg_cat = ADG_FFG
    test = False
```

Brand	Product name	Category	ADG	Prediction	Result
John West	Anchovies Fillets in Olive Oil	Pantry	1	1	True

Table 3. Rules Prediction example 1

Example #2

If Product Name contains ’Sourdough’ (upper or lower case), but the Product name does NOT contain 'biscuits', 'chocolate', 'snaps', 'olive', or 'cheese', then the product is ‘healthy’ (class 1). The first 2 rows in the table are class 1 (Table 4). The second 2 rows in the table are class 2. The second 2 rows were not assigned to the ‘healthy’ (class 1), despite having a ‘Sourdough’ in the product name. It happened because these two positions have ‘biscuits’ and ‘chocolate’ in their product names. Therefore, the last 2 rows assigned to class 2 (unhealthy).

```
if (re.search ("[Ss]ourdough", product_name) and
   (not re.search ("[Bb]iscuits", product_name)) and
   (not re.search ("[Cc]hocolate", product_name)) and
   (not re.search ("[Ss]naps", product_name)) and
   (not re.search ("[Oo]live", product_name)) and
   (not re.search ("[Cc]heese", product_name))):
    adg_cat = ADG_FFG
    test = False
```

Brand	Product name	Category	ADG	Prediction	Result
Coles Finest	**Sourdough** Baguette	Bread & Bakery	1	1	True
Coles Finest	Multigrain **Sourdough** Boule	Bread & Bakery	1	1	True
irrewarra	**sourdough** macadamia oat crunch biscuits	Pantry	2	2	True
irrewarra	**sourdough** chocolate & oat snaps	Pantry	2	2	True

Table 4. Rules Prediction example 2

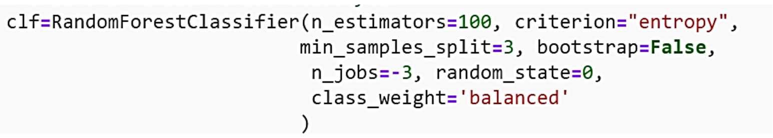
Following the classification, we performed an audit of rules to see the correctness of coding rules. Every rule was checked by the nutritionist using domain knowledge. The efficiency of every rule was checked on at least two observations.

5.2 Random Forest Classifier

Earlier we covered why we selected certain parameters for a Random Forest Classifier. During the implementation we made a few observations:

- **Criterion:** accuracy for class 7 with "entropy" slightly higher than the accuracy with default "gini".
- **min_samples_split:** min_samples_split=3, Better result than default 2
- **Bootstrap:** bootstrap=False, slight improvements for classes 1 and 7. For classes 3 and 8 improvements for some test runs.
- **N_jobs:** n_jobs=-3, more stable accuracy for class 7, on most runs there is an improvement for class 8.
- **Random_state:** We selected random_state=0 that used in every run, to get the same result.
- **class_weight:** class_weight='balanced', improved percentage of correctly identified samples from 44 to 56% for class 7.

The code for the RF model with previously describes the parameters:



```
clf=RandomForestClassifier(n_estimators=100, criterion="entropy",
                           min_samples_split=3, bootstrap=False,
                            n_jobs=-3, random_state=0,
                            class_weight='balanced'
                           )
```

Figure 2. RF classifier

The **Result** column (Table 5) assess the correctness of predictions:

Brand	Product name	Category	Energy	Protein	Fat	Sat Fat	Carbo Hydrate	Sugar	Sodium	ADG	Prediction	Result
nerada	organic green tea bags	Pantry	5	0.5	0.25	0.5	0.5	0.5	2.5	3	3	TRUE
Buderim Ginger	Ginger Refresher Cordial	Drinks	230	1	1	1	13.4	12.6	12	2	1	FALSE
Tassal	Premium Tasmanian Roasted Salmon Asian Style Sweet Chilli	Pantry	904	12.8	15.1	1.9	7.5	6.3	320	1	2	FALSE
kellogg's	coco pops	Pantry	1600	4.6	0.4	0.2	87.9	36.5	425	2	2	TRUE

Table 5. RF Predictions

The 'Result' column shows the correctness of the prediction. In the presented above 4 rows, we can see that product named 'Premium Tasmanian Roasted Salmon Asian Style Sweet Chilli' was identified as 'unhealthy' (class 2) product, when it is a 'healthy' product (class 1). The result column produced 'False' value for this observation.

Misclassification in the original data

In post checking process, we identified several misclassifications that were originally incorrectly identified manually. The Table 6 shows some examples. The most common misinterpretation is between class 1 and class 2.

Brand	Product name	Category	Energy	Protein	Fat	Sat Fat	Carbo Hydrate	Sugar	Sodium	ADG	Prediction	Result	Manual coding incorrect
continental	side dish teriyaki & garlic rice	Pantry	1520	8.5	1.2	0.3	79.5	6.7	750	2	1	FALSE	1
mamee chef	tom yam cup	Pantry	390	2.3	3.9	1.8	12	0.5	450	2	1	FALSE	1
white wings	cake mix caf style banana bread	Pantry	1470	6.1	16.6	2	44.2	20.8	445	1	2	FALSE	2
val verde	pasta sauce traditional	Pantry	143	1.6	0.2	0.7	6.4	4.6	490	1	2	FALSE	2
nestle	milo dipped snack bars	Pantry	1690	6.3	9.8	4.8	67.3	24.2	100	1	2	FALSE	2

Table 6. Misclassifications

6 Results

6.1 Rule-Base Classifier

The Rule-Based classifier is not fully complete for this research due to the limited time. Therefore, the analysis of its results by class is not possible. However, we can investigate the change of accuracy with different number of rules. We should remember that class 2 (unhealthy) set as a default class. It means that the accuracy with 0 rules represents the percentage of class 2 observations in the data.

Number of Rules	Percentage of correctly identified samples	Rules per 1% of accuracy increase
0	51.937262	-
100	55.930479	1200
145	60.915642	384

Table 7. Relationship between number of rules and accuracy

First 4% of increase in accuracy archived after 100 rules (Table 7). Next 5% of increase in accuracy archived after adding 45 more rules. In the first case, on average we need 25 rules to get 1% of accuracy improvement. In the second case, on average we need 9 rules to get 1% of accuracy improvement.

For this dataset, to achieve 100% of accuracy, we need to improve the original accuracy score by about 48%.

Using our previous estimations, we can assume that:

48*25=1200 rules 48*8=384 rules	The assumption is that to have a reliable Rule-Based classifier we need between 384 and 1200 rules.

But we also did a separate experiment on random set of 20 rules that gave us an improvement of 0.7%. It means about 28.5 rules for 1% of improvement in accuracy.

48*28.5=1368 rules	With this estimation we cannot deny a possibility that the classifier might require up to 1500 rules.

6.2 Random Forest Classifier

The Random Forest classifier trained on numerical data. The columns used for predictions are Energy, Protein, Fat, Saturated Fat, Carbohydrate, Sugar and Sodium. The overall accuracy: 0.924.

The misclassified percentage for every class performed by Random Forest classifier:

Class 1:	7.25 %
Class 2:	5.14 %
Class 3:	33.33 %
Class 7:	50.0 %
Class 8:	62.5 %

The accuracy percentage for every class performed by Random Forest classifier (Recall):

Class 1: 92.75 %
Class 2: 94.86 %
Class 3: 66.66 %
Class 7: 50.0 %
Class 8: 37.5 %

The percentage that our model correctly identified as True Positives (Recall) is higher for the **class 2** (94.86%). The next highest class (Table 8) by recall score is **class 1** (92.75%). C**lass 3** (Other foods, e.g., plain tea, coffee, baking powder, cream of tartar, food color, yeast, vinegar, herbs, spices) and **class 7** (special foods, e.g. alcohol, vitamins and supplements, baby food, infant formula). Class 8 (unable to categorize) accuracy should not be considered when assessing the effectiveness of the classifier.

	precision	**recall**	**f1-score**
1	0.91	0.93	0.92
2	0.94	0.95	0.94
3	0.87	0.67	0.75
7	0.89	0.50	0.64
8	1.00	0.38	0.55

Table 8. Accuracy Metric by class

The ratio of correctly predicted positive observations to all observations in actual class (recall) is the highest for class 8. The ration of correctly predicted positive observations (precision) to the total predicted positive observations is the highest for class 1. The highest weighted average of Precision and Recall (F1) is for class 1.

Some classes have limited number of data points. For classes 1 and 2 with the prevailing amount of data, the classifier identified correctly 93-95% of the products (Fig. 3). Smaller classes have lower accuracy. They contain categories like, baby food, alcohol, or coffee. This might happen because of limited data or because these categories based on more semantic meaning and classifier based on nutritional data is not able to take into consideration such nuances.

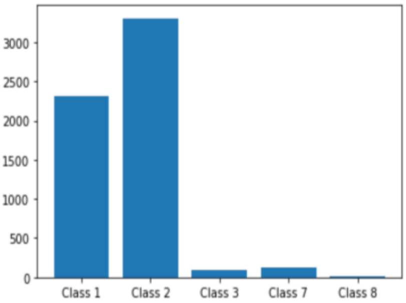
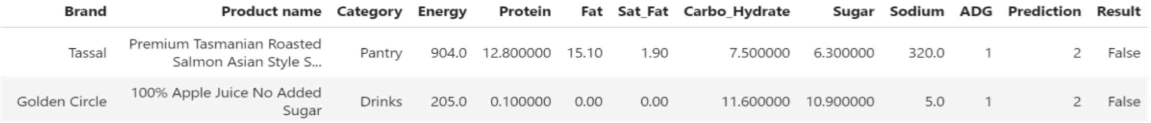


Figure 3. The ratio of classes

Random Forest accuracy by Class

Class 1 (Healthy Items)

The class 1 has 7% of misclassifications. We must assume that some of this may have been misclassified by human in the original data.

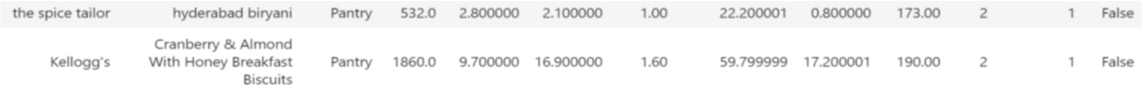


Brand	Product name	Category	Energy	Protein	Fat	Sat_Fat	Carbo_Hydrate	Sugar	Sodium	ADG	Prediction	Result
Tassal	Premium Tasmanian Roasted Salmon Asian Style S...	Pantry	904.0	12.800000	15.10	1.90	7.500000	6.300000	320.0	1	2	False
Golden Circle	100% Apple Juice No Added Sugar	Drinks	205.0	0.100000	0.00	0.00	11.600000	10.900000	5.0	1	2	False

Table 9. RF class 1 Predictions

Class 2 (Unhealthy Items)

The class 2 has 5.17% of misclassifications.

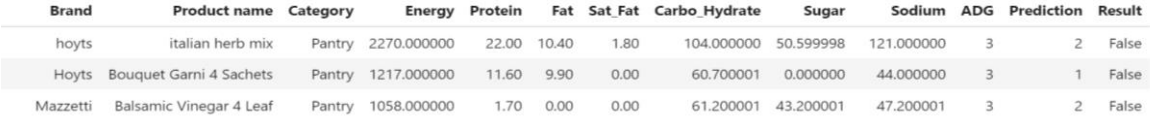


the spice tailor	hyderabad biryani	Pantry	532.0	2.800000	2.100000	1.00	22.200001	0.800000	173.00	2	1	False
Kellogg's	Cranberry & Almond With Honey Breakfast Biscuits	Pantry	1860.0	9.700000	16.900000	1.60	59.799999	17.200001	190.00	2	1	False

Table 10. RF class 2 Predictions

Class 3 (Other)

The class 3 mostly misclassified as class 2 (unhealthy) and class 1 (healthy). Items like 'tea' and 'coffee' were misclassified as class 1. This suggests that class 3 was assigned by nutritionist following the logic that could not be readable thought nutrition data.

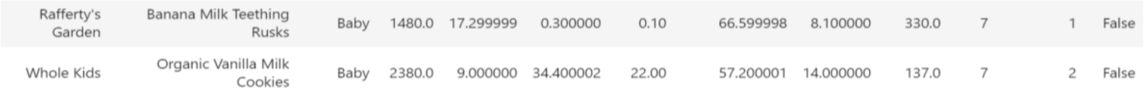


Brand	Product name	Category	Energy	Protein	Fat	Sat_Fat	Carbo_Hydrate	Sugar	Sodium	ADG	Prediction	Result
hoyts	italian herb mix	Pantry	2270.000000	22.00	10.40	1.80	104.000000	50.599998	121.000000	3	2	False
Hoyts	Bouquet Garni 4 Sachets	Pantry	1217.000000	11.60	9.90	0.00	60.700001	0.000000	44.000000	3	1	False
Mazzetti	Balsamic Vinegar 4 Leaf	Pantry	1058.000000	1.70	0.00	0.00	61.200001	43.200001	47.200001	3	2	False

Table 11. RF class 3 Predictions

Class 7 (Special Foods)

Class 7 (special food) was significantly misclassified (0.56) because this category contains numerous infant food products. These products difficult classified using nutrition data.

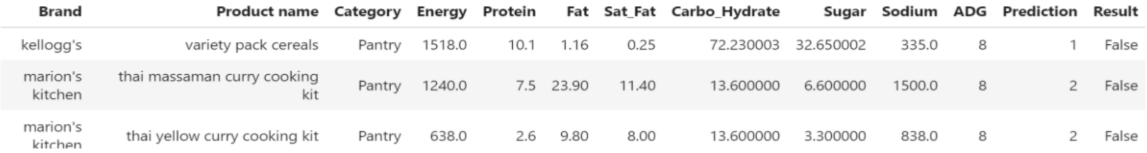


Rafferty's Garden	Banana Milk Teething Rusks	Baby	1480.0	17.299999	0.300000	0.10	66.599998	8.100000	330.0	7	1	False
Whole Kids	Organic Vanilla Milk Cookies	Baby	2380.0	9.000000	34.400002	22.00	57.200001	14.000000	137.0	7	2	False

Table 12. RF class 7 Predictions

Class 8 (unable to categorize)

Class 8 has a very limited number of samples. It does not follow any specific pattern and has only 38% of correctly identified observations. The most noticeable pattern for this class is the word 'kit' in the Product Name. Category like this could only be successfully identified by sentiment analysis or rule-based classifier.



Brand	Product name	Category	Energy	Protein	Fat	Sat_Fat	Carbo_Hydrate	Sugar	Sodium	ADG	Prediction	Result
kellogg's	variety pack cereals	Pantry	1518.0	10.1	1.16	0.25	72.230003	32.650002	335.0	8	1	False
marion's kitchen	thai massaman curry cooking kit	Pantry	1240.0	7.5	23.90	11.40	13.600000	6.600000	1500.0	8	2	False
marion's kitchen	thai yellow curry cooking kit	Pantry	638.0	2.6	9.80	8.00	13.600000	3.300000	838.0	8	2	False

Table 13. RF class 8 Predictions

7 Discussions and Limitations

The suggested solution has a potential to improve the efficiency to classify the healthiness of supermarket food and beverages.

Nutritional information can be used by researchers to classify groceries by the level of healthiness. The developed solution of **the Random Forest Classifier** can be used to classify products in the new grocery dataset. Such classification can be applied to any data without linking to country or language.

However, there are **some limitations** that revolve around the classification system that includes categories like 'baby food', 'tea and coffee', 'alcohol' or similar ones. It should be said that there is a high probability that the 'baby food' would be just classified as a 'healthy' category. While 'coffee' can fall into a variety of categories just by the difference in 'sugar' level.

The unbalanced classes create additional complication to reach a high accuracy. It means that some classes are difficult to predict because of the limited number of data points. Some of the misclassified items could possibly be incorrectly assigned in the original data. Overall, the classification accuracy could be improved, through model tuning and better data.

In addition, we raise a question, when we get a misclassification, is it possible that the classifier is more precise in its conclusions about some products than human who manually assign these categories?

The Rule-based classifier could be difficult to assess. The rules will only ever be as good as the "training set" we extracted them from. These rules might not be applicable for a different supermarket chain, in another country or in 5-10 years' time. Even if we create almost perfect set of rules for this dataset, eventually it may start misclassifying new products. The rules should be reviewed and updated as new products are added.

While The Random Forest has a potential to be highly accurate on any new data. The rules-based classifier use is limited by the location or time. However, we developed a proof of concept for rule-based classifier with under 100 rules. In the future with up to 1500 rules, such classifier can reach the accuracy of almost 100%.

8 Conclusion and Future Work

The Random Forest classifier reached an accuracy of 92.4%. We can say that it is possible to predict the 'healthiness' based on nutritional data. However, the highest accuracy for class 2 is 94.86%, and the lowest accuracy by class is 37.5% for class 8. In addition, the classification of products is not always responds to its nutritional qualities. In the future we can consider reviewing this solution by comparing different dataset with a variety of balanced and unbalanced classes.

After the reviewing the RF results, we found that some labels were assigned incorrectly in the original data, but the classifier was able to receive the correct class. For the future work we can replace those observations with the incorrect manual classification and repeat experiments using updated data. This will allow to slightly increase the RF classifier accuracy. We can also consider the implementation of other classification algorithms for numerical data. We believe that the RF classifier could still have a potential for improvement for class 1 and class 2.

The presented solution for Rule-Based classifier is "proof of concept". For Classifier to be efficient, many more rules are required. In the future work the restructuring of the rules should be performed.

The current accuracy that classifier was able to reach is about 61%. We did not assess the Rule-based classifier by classes, because we assigned rules without this attachment.

The Rules-based classifier establishes class 2 as a default class. If we consider a problem where we do not have a prevailing class, with the current number of rules, the accuracy will decrease significantly.

On average it took about 100 rules to increase accuracy by 5%. We can assume that to achieve accuracy of almost 100%, we will need between 384 and 1200 rules. However, to classify product in a real-life scenario we should take into consideration the limited use of some rules.

The random (less effective) 20 rules from this classifier added about 0.7% of accuracy. This leads to the conclusion that we might require up 1500 rules (the actual number can differ) in a real-life scenario to get close to 100% of accuracy.

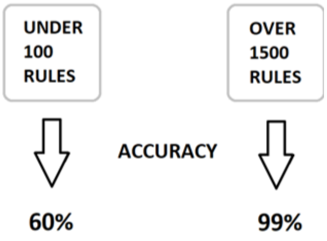


Figure 4. Accuracy Potential for Rule-Based Classifier

Overall, the Rule-based classifier showed more potential to be highly accurate in the future. To complete this research the Rule-based classifier should be supplemented with more rule. It would help to confirm or deny our suggestion that the classifier has a potential to reach about 99% accuracy. However, we should treat with caution the number of rules, because we might face a risk to have too many rules that would interact with only single item in the data.

References

1. Willis, O. (2019). Experts call for mandatory health star ratings on supermarket food. ABC News. [online] 1 May. Available at: https://www.abc.net.au/news/health/2019-05-01/processed-supermarket-foods-serious-health-obesity-problem/11061534.
2. The University of Sydney. (2017). 4 reasons why food is more important than you think. [online] Available at: https://www.sydney.edu.au/news-opinion/news/2017/10/16/4-reasons-why-food-is-more-important-than-you-think.html.
3. Healthstarrating.gov.au. (2019). About Health Star Ratings. [online] Available at: http://www.healthstarrating.gov.au/internet/healthstarrating/publishing.nsf/Content/About-health-stars.
4. Volodymyr Bilyk (2019). What is Sentiment Analysis? Definition, Types, Algorithms. [online] Theappsolutions.com. Available at: https://theappsolutions.com/blog/development/sentiment-analysis/.
5. Real Python (2019). Python "for" Loops (Definite Iteration). [online] Realpython.com. Available at: https://realpython.com/python-for-loop/.
6. Kuchling, A. (2019). Regular Expression HOWTO — Python 3.8.0 documentation. [online] Python.org. Available at: https://docs.python.org/3/howto/regex.html.
7. Ray, S. (2015). Python Regular Expression Tutorial | Python Regex Tutorial. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2015/06/regular-expression-python/#:~:text=What%20is%20Regular%20Expression%20and [Accessed 6 Oct. 2021].
8. thepythonguru.com. (2020). Python Regular Expression. [online] Available at: https://thepythonguru.com/python-regular-expression/ [Accessed 26 Sep. 2021].
9. Virmani, S. (2020). Rule-Based Classifier - Machine Learning. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/rule-based-classifier-machine-learning/.
10. Bizer (2020). Classification -Part 2. [online] Available at: https://www.uni-mannheim.de/media/Einrichtungen/dws/Files_Teaching/Data_Mining/FSS2020/DM02-Classification-2-FSS2020.pdf [Accessed 6 Oct. 2021].
11. Gavrilova, Y. (2020). How to Choose a Machine Learning Technique. [online] Serokell. Available at: https://serokell.io/blog/how-to-choose-ml-technique.
12. Brownlee, J. (2020). 4 Types of Classification Tasks in Machine Learning. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/types-of-classification-in-machine-learning/.
13. Wolff, R. (2020). Classification Algorithms in Machine Learning: How They Work. [online] MonkeyLearn Blog. Available at: https://monkeylearn.com/blog/classification-algorithms/.
14. Brownlee, J. (2020). Cost-Sensitive SVM for Imbalanced Classification. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/cost-sensitive-svm-for-imbalanced-classification/.
15. Yadav, P. (2018). Decision Tree in Machine Learning. [online] Medium. Available at: https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96.
16. Chen, C. and Liaw, A. (n.d.). Using Random Forest to Learn Imbalanced Data. [online] Available at: https://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf.
17. Tableau (n.d.). Data cleaning: The benefits and steps to creating and using clean data. [online] Tableau Software. Available at: https://www.tableau.com/learn/articles/what-is-data-cleaning.
18. Finxter. (2020). How to Split a String Between Numbers and Letters? | Finxter. [online] Available at: https://blog.finxter.com/how-to-split-a-string-between-numbers-and-letters/ [Accessed 6 Oct. 2021].
19. Sanad Zaki Rizvi, M. (2020). Applications Of Regular Expressions. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/01/4-applications-of-regular-

expressions-that-every-data-scientist-should-know-with-python-code/ [Accessed 6 Oct. 2021].

20. Tokuç, A.A. (2021). Splitting a Dataset into Train and Test Sets | Baeldung on Computer Science. [online] www.baeldung.com. Available at: https://www.baeldung.com/cs/train-test-datasets-ratio.
21. Goyal, C. (2021). Decision Trees Questions | Questions On Decision Trees. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/05/25-questions-to-test-your-skills-on-decision-trees/ [Accessed 26 Sep. 2021].
22. Fraj, M.B. (2017). In Depth: Parameter tuning for Random Forest. [online] Medium. Available at: https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d.
23. Koehrsen, W. (2018). An Implementation and Explanation of the Random Forest in Python. [online] Medium. Available at: https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76#:~:text=When%20training%2C%20each%20tree%20in [Accessed 26 Sep. 2021].
24. Albon, C. (2017). Handle Imbalanced Classes In Random Forest. [online] chrisalbon.com. Available at: https://chrisalbon.com/code/machine_learning/trees_and_forests/handle_imbalanced_classes_in_random_forests/ [Accessed 6 Oct. 2021].
25. Koo Ping Shung (2018). Accuracy, Precision, Recall or F1? [online] Towards Data Science. Available at: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9.
26. Molnar, C. (n.d.). 4.5 Decision Rules | Interpretable Machine Learning. [online] christophm.github.io. Available at: https://christophm.github.io/interpretable-ml-book/rules.html.