# Purchase prediction: what categories and brands within a category people will buy most likely?

Group 3: Alen Gabbassov, Vlad Miron, Fran Košutić, Michelle Mirchandani

# 1. Introduction

Everyday people around the world purchase different things for different reasons. One of the reasons people buy products is the brand's name. In every product category, there are brands that we are sticking to and brands that we trust and buy.

For many firms, the brand is among the most valuable intangible assets. Brand name alone can influence consumer perceptions and purchasing behaviour. For example, if we speak about buying computers, the majority of people will think about brands like Apple, HP, and Lenovo which account for more than 60% of the market share of the PC industry. (Canalysis.)

On the other hand, when people go shopping in retail stores there are a variety of complex quantitative as well as qualitative factors other than a brand name that the person needs to consider before making the decision. Personal promotions, dynamic pricing strategies, and historic purchase frequency are just a few decisions retailers, brands, and consumers face while interacting in this heterogeneous market of buying and selling goods.

In this research, we look exactly at the shopping baskets of the people in the supermarket to predict and prove that most of the time we make a decision not to buy the ***product*** but rather the ***category***. Furthermore, after making the decision of buying the category, there are a group of brands within a category that have the highest purchase frequency among all of the products of this category. Thus, our research question is: “What categories and brands within a category people will buy most likely?”. The proposed models for categorising the products and understanding category features offer practical value beyond purchase predictions. Retailers build models to improve their existing marketing decisions which affect business performance. Our model can predict the probability of brand level purchase probabilities unconditionally on category, which will help retailers and brands to observe and cooperate on relevant business decisions such as target advertising, a better understanding of purchase behaviour, and demand forecasting.

Section 2 presents the mathematical problem that we need to solve in order to answer our research question. Section 3 shows the data used and its characteristics. Section 4 discusses the approach we used in tackling this problem, including the features created and models employed. Then, we visualise our results in Section 5 and reach our conclusion and suggestions for future work in Section 6.

# 2. Problem Formalisation

In order to observe and understand which brands within categories are relevant, it is important to translate the business problem into a machine learning task using mathematical formalisation. In this report, the target variable and success metric that we want to predict is purchases (P). There are a number of customers (i) who enter a store, and time period, assuming it is in weeks (t), as shown in the dataset. Using vector of purchase indicators for customer (i) and week (t), we use

$b_{it} = [b_{it0}, \dots, b_{itJ}] \in \{0,1\}^{JX1}$ to distinguish customer purchase decisions of each customer (i) at a time period (t). This leads us to having a binary purchase indicator for customer (i), week (t), and product (j) $b_{itj} = \in \{0,1\}$, which indicates whether the customer purchased the product at a certain time.

Using purchase frequency of length (T) for customer (i) at time (t) $B_{it}^{T} = [b_{i,t}, b_{i,t-1}, \dots, b_{i,t-T+1}] \in \{0,1\}^{JXT}$, we can create a vector of product-specific purchase indicators using over the entire time period of length (T) using$B_{it}^{\infty} = [\bar{b}_{it1}, \dots, \bar{b}_{itJ}] \in \{0,1\}^{JXL}$. As mentioned in the introduction, customers receive personalised coupons, denoted by $D_{it} = [d_{it1}, \dots, d_{itJ}] \in \{0,1\}^{JX1}$, where size of coupon discount received by the customer (i) at a time (t) for product (j) is denoted by $d_{itj} = \in \{0,1\}$.

To predict purchase probability ($p_{i,t+1} = [p_{i,t+1,1}, \dots, p_{i,t+1,J}]$ ) that customer (i) purchases product (j) at time (t+1) for every product ($j \in \{1, \dots, J\}$), coupon assignment ($D_{i,t+1}$), purchasing history ($B_{it}^{T}$), purchase frequency ($B_{it}^{\infty}$) and model parameters θ, we have the learning task: $P_{i,t+1} = f(, B_{it}^{T}, B_{it}^{\infty}; \theta)$. The vector of purchase probabilities of (i,t) $P_{i,t+1}$ contains the probabilities for binary purchase events of all products (j) where customer (i)'s purchase probability for i, t and j is ($p_{i,t+1} = \mathbb{P}[b_{i,t+1,j} = 1]$). Separating the information directly during the model input was conducted to make the learning process and training more efficient. Providing both $B_{it}^{T}$ and $B_{it}^{\infty}$ reduces the dimensionality of the data. Since recent purchases are more relevant in the model, we included $B_{it}^{\infty}$ as a summary of older purchases and considering a smaller window T. The model assumes cross-product relationships from the transaction data.

We tackle the problem using supervised learning. In our case, we have the labelled input for our raw data in baskets-s.parquet as well as in coupons-s.parquet. In the next sections, we would explain the choice of the models and explore the data in order to give a better outlook on forming the categories and understanding the probabilities of the brands within the categories.

# 3. Data

The data used to help us answer the research questions are datasets baskets-s.parquet and coupons-s.parquet. The baskets-s dataset contains products which each customer that visited the store that week purchased. The dataset contains four columns - week, customer, product, and price. As the raw dataset is already labeled, using supervised learning is more appropriate. Each customer has their own basket, which in turn has many products, each with their unique ID. Each product also has the price it was purchased for. This dataset is quite large, with more than 68 million entries. The coupons-s dataset contains the products that the customer received coupons on each week. The dataset also has four columns - week, customer, product, and discount. Each customer received coupons for various products each week. Each product also has the discount in percentage terms listed. Similarly to the baskets-s dataset, coupons-s is quite extensive, at more than 44 million entries. There are in total 90 weeks, and our model needs to predict purchase probabilities for the 90th week. The other 89 weeks will be used for training and validation. In total, there are 250 different products purchased by 2000 customers in the 89 weeks. More details regarding the distribution of the raw data are shown in ***Table 1.***

| | **Week** | **Customer** | **Product** | **Price** |
|---|---|---|---|---|
| **Count** | 1378720.0 | 1378720.0 | 1378720.0 | 1378720.0 |
| **Mean** | 44.5 | 997.2 | 125.1 | 584.3 |
| **Std. dev.** | 26.0 | 576.3 | 69.7 | 97.4 |
| **Min** | 0.0 | 0.0 | 0.0 | 0.0 |
| **25%** | 22.0 | 499.0 | 66.0 | 506.0 |
| **50%** | 44.0 | 993.0 | 123.0 | 579.0 |
| **75%** | 67.0 | 1496.0 | 189.0 | 654.0 |
| **Max** | 89.0 | 1999.0 | 249.0 | 837.0 |

***Table 1.*** *- Distribution of raw data*

When building our prediction model, we need to be aware of the efficiency, otherwise, the large amounts of data can impair our ability to progress effectively. As our research question concerns brand categories, we will first be creating a correlation matrix of products purchased (See section 5.1). From the matrix, we will be able to see which products are most likely substitutes or complementary products, from which we will be able to deduce product categories.

To calculate baskets, we group products purchased by customers and week. You can find the GitHub link for the code attached in the Appendix. The average number of products per

basket is 7.66. The minimum number of products is 1 and the maximum is 20. The standard deviation is 2.38. A visual representation is shown in **Figure 1.**

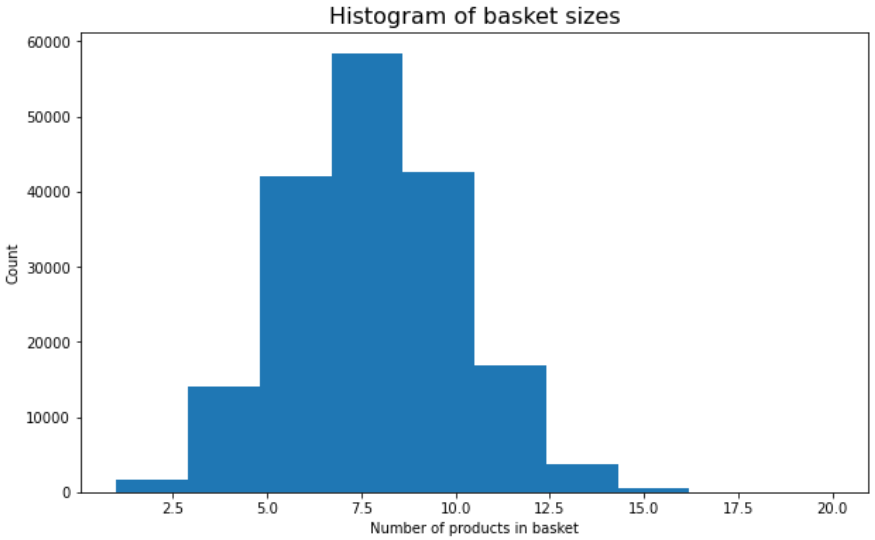


***Figure 1.*** *- Histogram of basket sizes*

Another interesting metric that could influence our predictions for week 90 is the number of items sold per week. If there is a certain pattern in how the number of items sold per week fluctuates, we could incorporate that into our prediction models. At first glance at **Figure 2.**, one might think that there is a lot of variance in quantity sold each week at the store. However, the values range from about 15045 to 15605, which is only about a 4% gap. Furthermore, the mean is 15319 and the standard deviation is only around 98, hence we are not expecting a large movement in an overall number of items sold in week 90.

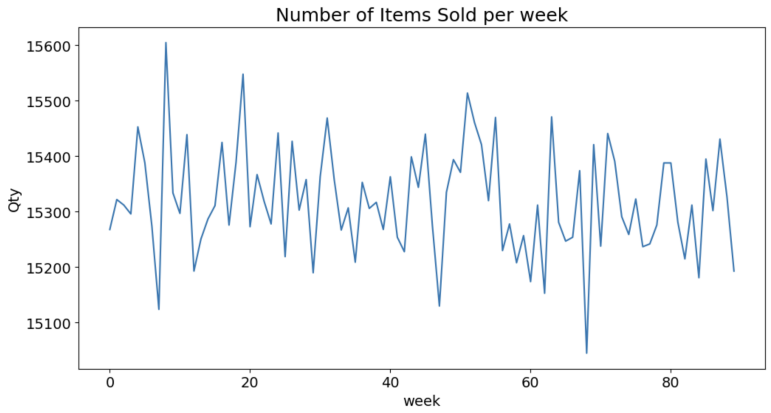


***Figure 2****. - Quantity of items sold each week*

As customers might not be loyal to one brand from a certain category, they will still likely purchase products from the category on a regular basis. For instance, customers every month or so will likely

buy a toothbrush. As the toothbrush category is quite commoditized and firms often compete mostly on price as the functionality of the majority of the products is the same. Even when the functionality differs, customers are likely to not stick to a particular brand (at least in the beginning) as they will likely want to experiment with multiple brands to see which one fits their preferences most. So, along with trying to predict which specific toothbrush brand the customer will most likely purchase, we want to also analyse whether the customer will need a toothbrush, only predicting the purchase of specific brands later.

Furthermore, we also have data on coupons. Specifically, for which products coupons were received, how much did they discount the price, and in which week were they received. Coupons likely have an impact on purchase probabilities of customers, as in customers are more likely to buy discounted brands over those they usually buy. Coupons will also have effects on the purchase probabilities of categories. For instance, if a customer buys nacho chips, and sees that a salsa dip is discounted, they are more likely to buy it even if originally they did not plan on it. To account for this effect in the creation of our model, the discounted products purchased will have less weight in predicting future purchases of a category or a specific brand. Furthermore, coupons can also impact purchase decisions as customers might delay their purchases of certain items in anticipation of receiving a coupon in the future. This effect will mostly be present in items that are not bought on a frequent basis, such as cleaning products that last for several months after purchase. As customers notice they are almost running out of a certain cleaning product, they might start looking out for the coupon and only then purchase that product. However, this effect will almost be non-existent for products that are purchased frequently, such as bread, as customers do not have time to wait for a coupon.

# 4. Approach

We approach our research question using 3 models applied on the same explanatory variables, predicting the same target of product purchase probabilities for week 90. Subsection 4.1. discusses the features we build with the intention of using them in our models, while Subsection 4.2. shows the models employed on the built features.

## 4.1. Features Engineering

One of the three explanatory variables is the frequency of purchase for every given product. This is calculated as the number of times a product was bought by all customers over the period from week 1 to week 88 divided by the number of baskets in this same period. This feature takes advantage of the human tendency to choose products and services which they are familiar with, products which instil trust. We expect that a purchase of a given product in the 88-week time window will increase the

probability that the customer will buy that same product in week 89 and 90. On the flipside, one downside of this feature is that perhaps products that were bought a long time ago might have either been forgotten or not bought on purpose, due to offering an unpleasant experience in the past.

The second feature is created similarly to the first one, but only taking into account the most recent 30 weeks. This may improve upon the issue of including purchases of products which the customer is no longer interested in, as products which have not been purchased in the most recent 30 weeks are not assessed.

The third feature is similar to the first and the second but only takes into account the most recent 5 weeks. This may give different insight compared to the first two, as we expect a purchase in the past to increase the probability of a purchase of the same product in the future, except for the short period following immediately after the initial purchase. At that point, we think that the propensity to buy the same product may actually decrease since the customer hasn't had the chance to consume the product and wish to have another one.

## 4.2. Models Used

### 4.2.1. Logistic Regression

Logistic regression is our baseline model. It represents a good starting point as it is inherently more interpretable than random forest and boosted trees, which tend to become black boxes given the high tree depth and the number of leaves they create. With logistic regression, the intercept and the coefficients may indicate the impact of our independent variables, which we may then check against our intuition. Therefore, logistic regression is our baseline model which we use to understand the impact of the features used and assess the performance of the other models.

### 4.2.2. Random Forest

[Ref.1] states that Random forest (RF) is one of the tree ensemble methods that train multiple regression trees using bootstrap subsamples of the data points and features. This model processes every sample of the created trees and forecasts the target variable by averaging the forecast outcomes from multiple trees. [Ref.1] further shows that RF can solve the overfitting problem, which is a typical issue when a single decision tree or regression tree analyzes a complex dataset and hence produces an accurate forecast. More specifically, the bootstrap sampling process divides a single giant and deep tree into several small trees in which overfitting is less likely to occur. [Ref.1] proves that within each tree, this algorithm uses the mean squared error (MSE) to calculate the deviation of the prediction from the target value. Also, RF uses a splitting rule to find the best partition that minimises MSE. However, the advantage of obtaining an accurate forecast comes at a cost of interpretability. [Ref.2]

points out that as RF utilises multiple trees to calculate its forecast, it is less intuitive to understand the decision-making process compared to a single regression tree. A higher number of leaves implies less interpretability of the model. This work uses a standard library in the machine learning package sklearn in Python to implement RF.

### 4.2.3. Boosted Trees

The boosting of the model is implemented by using Decision Trees (DT). In contrast to RF, [Ref.3] points out that Boosted Trees (BT) is a method that works by combining many weak DTs, most of which are usually tree stumps (a decision tree made out of only one node and two leaves). A stump produces relatively inaccurate rules as it can only use one variable to make a decision. Then, using a forest that is made out of (mostly) stumps, BT goes on to classify the samples. An important distinction that is made between RF and BT is that BT attributes different weights to this forest of stumps, whereas RF attributes equal weights to all DTs in the forest. Also, each stump is made by accounting for the previous stump's shortcomings, making BT impossible to parallelize, as the order of the generated trees matters (whereas in RF each tree is generated independent of the other trees).

More precisely, BT first builds stumps for each attribute in the dataset and then selects the stump which has the smallest GINI index, as the DT of said attribute correctly classifies the most observations in the sample. Then, [Ref.4] shows that ADA has to determine the *OutputWeight*, which is the weight attributed to the stump, via the following formula:

$$OutputWeight = \frac{1}{2}\log\left(\frac{1 - Total\ Error}{Total\ Error}\right)$$

will be a large negative value if the stump consistently predicts the opposite classification of the correct one, and a large positive value if it consistently correctly classifies. [Ref.3] suggests that BT has to use the above-mentioned *OutputWeight* to resample the original dataset. This is done using the following formula for the misclassified samples:

$$UpdatedSampleWeight = PreviousSampleWeight \times e^{OutputWeight}$$

is large (which means the previous DT did a good job in classifying most samples), the *PreviousSampleWeight* is scaled by a large value, since that implies the observation needs to be especially taken into account as the DT used for it was generally an accurate one, yet it misclassified it. The weights of the samples which are correctly classified are scaled via a similar formula to that of the misclassified samples, except that *OutputWeight* is multiplied by *-1*. The minus sign in front of the *OutputWeight* ensures that correctly classified sample weights are going to be decreased proportionally to how accurate the DT was. This way, samples that were misclassified will gain in weight and samples correctly classified will decrease in weight. This will ensure that the following

stump will be additionally incentivized to correctly classify the previously misclassified observations. [Ref.4] states that the next step is normalising the sample weights, such that they sum up to 1. We use the normalised weights to create the second stump and iteratively repeat this process. Finally, we sum up the weights of all stumps in the forest and use the outcome of the group of stumps with the largest weight as the prediction of BT. This model is implemented using a standard library in the sklearn machine learning package in Python.

# 5. Results

We use 3 models and 3 explanatory variables to predict the purchase probability in week 90. They are logistic regression (the baseline model), random forest, and boosted trees. All of them use the same explanatory variables and have the same target. The explanatory variables are the purchase frequencies taken over the entire training period, the most recent 30 weeks and the most recent 5 weeks.

## 5.1 Co-occurrence matrix

The co-occurrence matrix shown in **Figure 3** shows how purchases of one product are correlated with the purchases of another product in the dataset. The brighter the colour, the higher the number of times the 2 products (on the x and the y-axis) co-occurred in the same basket. With completely random data, one would expect that these co-occurrences would show randomly in the matrix, creating no clear blocks of complementary purchases. However, as seen in **Figure 3**, the co-occurrences follow certain patterns. This points to the product data not being completely random. Likely the data contains certain categories of products which are bought together, producing such co-occurrence structure.

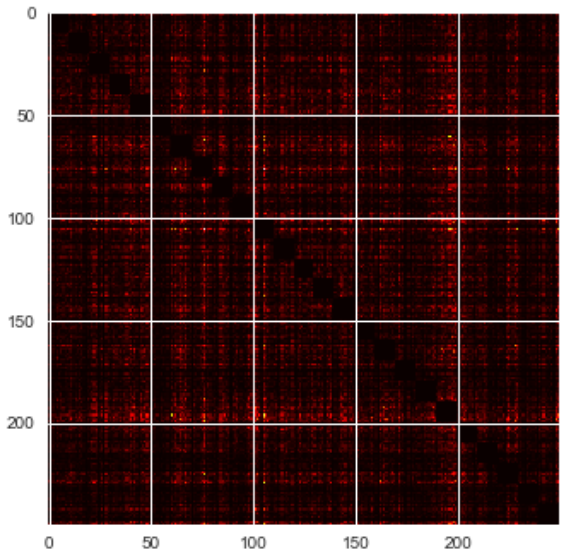

***Figure 3. -*** *Co-occurrence matrix*

### 5.1.1 Leaders within Categories.

As stated in our research question we want to explore brands within a category and investigate whether there are products which lead in the used categories. We have 250 products which are grouped into 25 categories. Category-leading products are defined as products which have a higher than 10% probability of being purchased given that the category they are part of is chosen. An over 10% probability is disproportionately high as we would expect roughly 10% purchase probability for all products in a category if they were a perfect substitute for each other and customers had no preference for any of them. Every customer has its own product leaders within each category which is why we take an average over the product purchasing probabilities of all customers over all weeks.

The mean conditional probabilities of purchasing a product given we purchase within its category are shown in Appendix 2. As seen from the table, some categories have a few leaders which make up roughly all the sales within the 10-product category, while some categories have no leaders, having a roughly uniform distribution of sales among the 10 products that make up said category.

This derived output is interesting since it might lead to new insights. Some retailers might want to look into those categories where there is no clear market leader or people just don't have a preference for any product to disrupt said categories and gain market share. At the same time, categories such as category 19 where only 5 products make up roughly 70% of all sales may be an interesting case-study to see how these 5 brands positioned and marketed themselves to become such outstanding leaders.

## 5.2 Training Results

The logistic regression results show that purchase of a product decreases the probability of repurchase within a 5-week window, while it increases the probability of repurchase in a 30-week or entire timeline window. This can be observed by the high, 10.747 coefficient for the frequency of purchase over the entire timeline, indicating that a product once purchased is an important predictor for the repurchase of the same product, given a high enough time window. This repurchasing of products which were previously purchased is also present when looking at a 30-week window, although with a value of 0.308, we see that it has a much smaller effect. This indicates that many customers are not ready to repurchase the same product after only 30 weeks. The opposite effect is present on a 5-week timeline, where we see a negative coefficient of -0.981, showing that many customers are not ready to repurchase a product after only 5 weeks, meaning they likely did not finish using the previously bought good or the desire of repurchase has not built up fast enough.

| | Logistic Regression | Random Forest Training | Boosted Trees Training |
|---|---|---|---|
| **MAE** | 0.043 | 0.041 | 0.043 |
| **MSE** | 0.023 | 0.021 | 0.021 |
| **R2** | 0.229 | 0.308 | 0.276 |
| **Log Loss** | 0.090 | 0.075 | 0.078 |
| **Intercept** | -4.485 | | |
| **Coefficient** | 10.747, 0.308, -0.981 | | |

***Table 2.*** *- Model performance on Training data*

According to the results of the logistic regression, the r-squared value is 0.229, suggesting that 22.9% of purchase behaviour can be explained by the 3 independent variables used. This value (lower than 50%) indicates that the logistic regression model's independent variables do not explain the variability of the probability of repurchase well.

Compared to the r-squared figures of the random forest and boosted trees training, the logistic regression performed the worst, while the random forest training performed the best.

$$Y = -4.485 + 10.747x_1 + 0.308_2 - 0.981x_3$$

According to the logistic regression's Mean Absolute Error (MAE), the average of the absolute difference between the actual and predicted values in the dataset is 0.043. The MAE measures the average of the residuals in the dataset. Furthermore, the logistic regression's Mean Squared Error (MSE) is 0.023. The MSE represents the average of the squared difference between the original and predicted values in the data set. The MSE measures the variance of the residuals. A lower value of MAE and MSE implies a higher accuracy of the regression model. MSE penalises large prediction errors (outliers) more than MAE. Using these results, we can tell that the logistic regression has a slightly lower accuracy than random forest and boosted trees on the training sample.

Log-loss is indicative of how close the prediction probability is to the corresponding actual/true value. A lower log loss value means better predictions. The random forest training is the lowest log loss value (0.075), meaning that it is the closest to the actual/true value.

Thus, based on the training sample we prefer RF as it achieves the highest R-squared (30.8%), lowest values for the error measures (0.041 for the MAE, 0.021 for the MSE) and lowest log-loss value (0.075).

## 5.3 Validation Results

| | **Log. Regression** | **Random Forest Training** | **Boosted Trees Training** |
|---|---|---|---|
| **MAE** | 0.043 | 0.043 | 0.043 |
| **MSE** | 0.023 | 0.022 | 0.021 |
| **R2** | 0.226 | 0.245 | 0.271 |
| **Log Loss** | 0.090 | 0.088 | 0.077 |

***Table 3.*** *- Model performance on Validation data*

According to the test model performance of week 89 (Table 3), all three models performed similarly in terms of MAE and MSE, with a slight edge taken by Boosted Trees due to the best MSE performance. Furthermore, boosted trees training scored the best in terms of r-squared (27.1% of purchase behaviour can be explained by the independent variables). Lastly, boosted trees training had the lowest log loss.

We do not prefer logistic regression and random forest due to the underwhelming performance in the validation set compared to boosted trees. Furthermore, random forest shows signs of overfitting on the training set, having a large drop in terms of R-squared and log-loss in the validation set. Boosted trees achieve the best results in the validation set and are relatively close to the performance it had in the training set. Showing no signs of overfitting and achieving the best results, we choose boosted trees as the best model to answer our research question.

# 6. Conclusion and Next Steps

Using past purchase frequencies over 88, 30, and 5 weeks, we used machine learning algorithms to make predictions of purchase probabilities in week 90 for each product within the dataset. We have also analysed the purchase probability of categories and subsequent conditional probabilities of choosing brands given that we choose their category. As such, we wanted to analyse the human tendency of sticking with brands they are used to within a category.

As seen from the results shown in Table 2 and Table 3, our models generally have a R-squared value which tends to 30%. This indicates that although the 3 explanatory variables used explain part of the purchase probability, there are other factors at play which may be omitted from these models. In terms of accuracy measures we analyse the MAE and MSE. Our predictions range between 0 and 1. As such, a 0.04 mean absolute error can prove to be quite high in some cases. However, the MSE for our models ranges between 0.021 and 0.023, which is significantly less than the MAE and points to predictions not ranging from the mean as much. Thus, our models generally have a low log-loss value, achieving values under the 0.1 threshold across all models and both the training and validation sets.

On the training data, random forest performed best, showing the lowest MAE, MSE, highest R-squared and best log-loss. That said, it seems to show overfitting as its performance drastically decreases across all measures once we move onto the validation set. Boosted trees showed lower performance in the training set, but stayed consistent in the validation set and outperformed random forest and logistic regression. Due to having the best performance in the validation set and showing consistently good performance compared to the training set, we choose boosted trees as our preferred model to predict the purchase probabilities of the products in week 90 and answer the research question.

Regarding future research opportunities it would be insightful to use the large, "l", dataset for the research, especially as it concerns plotting the product co-occurrence matrix. Currently there are some patterns which are showing in the "s" and "m" datasets, but there is still a more clear structure which may be revealed by analysing the "l" dataset, leading to categories which are more homogeneous (products are better complements of each other). Furthermore, due to the data being clean and having many issues such as missing data or additive outliers being removed, it may be hard to use the presented models in the real world. Future researchers may start the modelling from raw retailer data, having to create models and processes to clean the data first, having better results in the real world and potentially creating models that may generalise to other industries.

# 7. References

[Ref.1] - Breiman, L. (2001). Random forests. Machine learning, 45(1):5–32.

[Ref.2] - Liu, S., Dissanayake, S., Patel, S., Dang, X., Mlsna, T., Chen, Y., and Wilkins, D. (2013). Rule based regression and feature selection for biological data. In 2013 IEEE International Conference on Bioinformatics and Biomedicine, pages 446–451. IEEE

[Ref.3] - Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1):119–139.

[Ref.4] - Schapire, R. E. (2013). Explaining adaboost. In Empirical inference, pages 37–52. Springer.

# 8. APPENDIX 1:

Github link for the code: https://github.com/VladMiron00/Recommender-System

# 9. APPENDIX 2: Product leaders within category

| Category | Product | Mean Conditional Probability |
|---|---|---|
| **2** | 22 | 0.105401 |
| **4** | 40 | 0.125699 |
| **6** | 60 | 0.148584 |
| **6** | 67 | 0.110737 |
| **7** | 76 | 0.164883 |
| **9** | 98 | 0.111077 |
| **10** | 101 | 0.125821 |
| **10** | 105 | 0.152243 |
| **13** | 138 | 0.100531 |
| **19** | 192 | 0.116105 |
| **19** | 196 | 0.161793 |
| **19** | 197 | 0.126079 |
| **19** | 198 | 0.151640 |
| **19** | 199 | 0.143125 |
| **22** | 225 | 0.104752 |
| **22** | 229 | 0.112607 |