A Work Project, presented as part of the requirements for the Award of a Master's degree in Business Analytics from the Nova School of Business and Economics.

Developing a dynamic recommendation system for the aftermarket car industry:

Dimensionality Reduction techniques to improve recommendation system

The TIPS 4Y use case

João Maria de Sá Nogueira Almeida Colaço - 44661

Polina Masalovich - 46099

Work project carried out under the supervision of Prof. Carlos Santos

17-12-2021

ABSTRACT:

Recommendation systems are a powerful tool for e-commerce businesses when applied properly. After extensive research on the various models used in the market, a recommendation system was successfully designed, with the purpose of helping TIPS 4Y support after-market distributors by optimizing the recommendation of auto parts. Adopting a CRIPS-DM framework, an Item-Based Collaborative Filtering was developed, with the data provided by TIPS 4Y, that performed far better than the other two baseline models. Attempts to improve the model, Dimensionality Reduction techniques were tested. A data dashboard was also designed, allowing TIPS 4Y to make management decisions on working with data.

Keywords: Business Analytics; Data Science; Recommendation System; TIPS 4Y; Automotive Aftermarket; Dimensionality Reduction; Data Dashboard.

Acknowledgments: Our team would like to thank our advisor, Professor Carlos Santos, for all the support provided. We are also truly grateful to our mentors Susana Neves and Mariana Henriques, that without their encouragement, this thesis would not have been possible. A word of thanks to the entire Project Based Learning organizers, particularly Professor Lenia Mestrinho, who gave us a unique chance to work on a real-world project, and Eduardo Garcia, who shared his extremely helpful computational skills. Finally, we would like to thank TIPS 4Y, especially Allan Souza and Miguel Cabral, for trusting us with this project and for their constant availability and patience.

INDEX

1. BUSINESS UNDERSTANDING

1.1 The industry

The automotive industry is one of the most dominant sectors of the Portuguese economy. It is currently the 3rd largest in the scope of the manufacturing industry and accounts for 19% of the national GDP, 25% of the exports of tradable goods and directly employs 200 thousand workers. At the same time, the automotive market is not limited only to the production and sale of cars but also to various kinds of automation.

1.2 The company

TIPS 4Y is one of the only Portuguese companies that provides information systems for the automotive industry. The company was founded in Lisbon in 2012 and offers "innovative solutions with an impact on mobility and user experience across the auto ecosystem." (TIPS 4Y's website. They offer a variety of products with '*Catálogo Tec Doc*', '*Pesquisa por matrícula*' '*Webshop TecDoc'* and '*DataDrive'* being their most attractive ones. The *Webshop* is a tool that enables users to digitize their business customers purchasing process. The findings of this thesis will directly impact the performance of this tool.

TIPS 4Y's offers are tailored to the various types and sizes of organizations in the aftermarket car industry. The company's customers are divided into three groups. There are the Distributors who are entities that use TIPS 4Y's *Webshop* and make it available for retailers and in some cases directly to car repair shops. Then there are the Retailers, companies that buy parts from distributors through their *Webshop* then sell to consumers. Finally, car repair shops are end-users for retailers and distributors who make orders using the TIPS 4Y *Webshop* solution.

1.3 Organizational challenge to be addressed

TIPS 4Y believes they are losing possible revenue due to lack of information on its customers purchasing habits and a tool to benefit from that information. The management of the company trusts that the problems could be solved through implementation of a recommendation system and management data dashboard. In other words, if there was a system in place that showed articles that are frequently bought together, that the client will most likely need given the items he is purchasing. Hence, the company has the following challenges:

- TIPS 4Y 's clients miss out on revenue due to no recommendation engine implemented in the *Webshop* and therefore, miss out on cross-sell opportunities.
- Warehouse owners lack information on which vehicles need what services and spare parts and thus, organize their inventory inefficiently.

2. LITERATURE REVIEW

A literature review was conducted to gain the appropriate knowledge for the methodology and to help develop this project. Various experts were consulted which led to different frameworks and ideas creating the foundations for this thesis.

Given the nature of this thesis, a Data Science project, an appropriate process framework should be adopted. Of all the various frameworks available, CRISP-DM is the one that stands out. It is an effort to provide a standard framework for industrial projects which is composed of 6 phases: Business understanding, data understanding, data preparation, modelling, evaluation, and deployment. This framework has become the most popular amongst the data science community. Nadali reported that 42% of companies have adopted this CRISP-DM in their projects (Nadali et. al. 2011).

Recommendation systems are one of the most popular data mining and machine learning applications in the internet business. The recommendation system analyzes the behavior of users of the Internet service, after which it can assess the user's preference for a particular recommendation object. The objects of recommendations can be products in the online store, a set of sections of the website, media content, and other users of the web service. Based on the preferences predicted by the recommender system, the behavior of the web service to a specific user can change - this functionality is commonly called personalization. There have been recent developments in the various techniques for building a recommendation system: **Content-Based Filtering (CBF)**, **Hybrid Filtering**, and **Collaborative Filtering (CF)**.

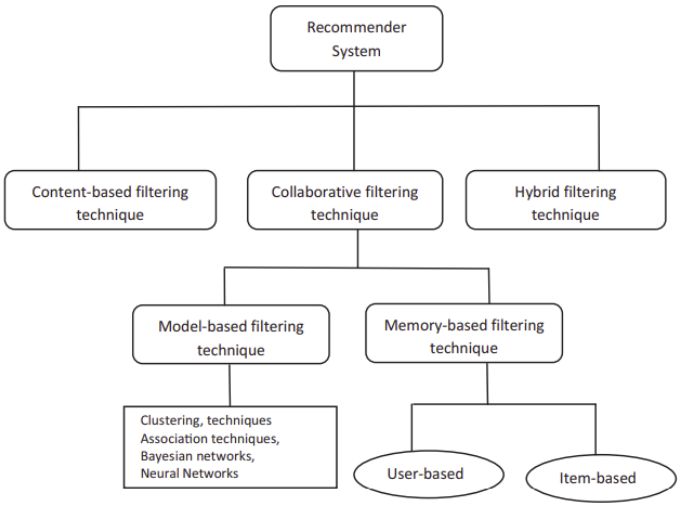

Figure 1 - Types of Recommendation Systems

Content-based filtering focuses more on analyzing the characteristics of the items to make the recommendations. CBF technique is built on user profiles, using characteristics taken from the content of the items the user has previously rated. Thus, the recommendations made by this technique are the items that are most similar to the ones most favored by the user. That being said, CBF is most successful when the items being recommended are web pages, news, publications etc, meaning items that have content in itself. To calculate the similarities this technique uses different

types of models ranging from Probabilistic models, such as Neural Networks, Decision Trees, and Naïve Bayes Classifier, to Vector Space models, like the Term Frequency Inverse Document Frequency. Some of the advantages of CBF are that the recommendations are not influenced by other user-profiles and that it can adapt the recommendations in a short period of time if the profile of the user changes. The biggest downside of this technique is that it is dependent on a strong understanding of the content of each item (Isinkaye et al. 2015 p. 264-265).

Hybrid Filtering combines different recommendation techniques so that it can have a greater system optimization, to avoid constraints and shortcomings that pure recommendation techniques have. The basis of this technique is that using a combination of various algorithms will give out better recommendations rather than only using one since the disadvantages of one algorithm can be covered by another one being used in the combination. The approach of combining the different techniques can be done in various ways like applying the algorithms separately and combining the results or doing a mixture between CF and CBF approach. The major disadvantage of this technique is that it generally has a very high computational complexity and needs a substantial amount of data that is up to date, or it will be hard to retrain the model and keep providing new and accurate recommendations (Vatsal 2021).

Collaborative Filtering is the most used technique whether for commercial or academic purposes. The idea is the process of finding the interests of a user by detecting information and preferences from various users. This is achieved by filtering data for patterns and useful information by using methods involving data sources, a collaboration between multiple agents. In simple terms, the basic idea is that if users X and Y share similar preferences towards an item, then X and Y are likely to share similar preferences towards other products. CF is divided into two common approaches, model-based and memory-based (Luo 2018).

Model-based uses predictive machine learning models like decision trees, association approaches etc. Features related to the data are parameterized as inputs for the model to work out the optimization problem.

Memory-based is normally divided into user-based and item-based. Hahsler in 2011, stated that User-based seeks to simulate word-of-mouth based on the analysis of rating data. The idea is that users with similar tastes will rate items the same way, meaning, ratings can be predicted by detecting a neighborhood of similar users to then combine those ratings to build the recommendation. The neighborhood is detected by k nearest neighbors or by all users that are under a similarity threshold defined a priori. The most used similarity measures are the Cosine similarity or the Pearson correlation coefficient. Item-based on the other hand generates recommendations by calculating an item-to-item similarity matrix using the same similarity measures as the user-based approach. The idea is that users have a higher preference towards items that are similar to the items they had previously rated (Hahsler 2011 p.3).

Item-based CF is used by many 'tech giants', Amazon being one of them. However, given the massive volume of data that they operate every day, they had to adapt and improve the model in order to produce real time, high-quality recommendations. Their systems allow customers '*filter their recommendations by product line and subject area, rate the recommended products, rate their previous purchases, and see why items are recommended'* (Linden 2003 p. 76–80).

3. METHODOLOGY AND ANALYSIS

As mentioned in the literature review, the methodology used for this thesis will be based on the CRISP-DM framework. The first step, Business Understanding, has already been tackled in the 'Organizational challenge to be addressed' section of this thesis.

The next step is Data Understanding which is the first analysis of any data science project. In this phase, answers must be found for the following questions in order to understand how to proceed with the project: What data is at our disposal (main data characteristics)? What is the level of data quality (duplicate records, input errors, etc.)? If there is not enough data, how to transform the data for the next steps? In case of insufficient data, what additional data is required?

During this phase of the project, it was decided to proceed with the data preparation simultaneously.

3.1 Data Understanding

The company TIPS 4Y has given us access to a year's sample of their data, more specifically, from the 30th of April 2019 to the 4th of September 2020. It accounts for 590 318 different orders from 723 different clients. With this data will be developed a dynamic recommendation system and data dashboard.

Starting by analyzing the database that was provided by TIPS 4Y, it was clear that a data dictionary had to be built (see Appendix A) to understand the overall composition of the database and help maneuver the data more easily.

The next step was to also decide from within these tables which variables are relevant to the project and did not contain missing values. Some variables were excluded as they would not add any value to the future model and dashboard, or the data was incomplete. At the same time, some variables needed to be transformed to the proper data type, for example from string to integer.

Hence, after reviewing the data with the client ended up were chosen 6 tables and 19 variables (see Appendix B.)

After the first analysis, to ensure that the data was coherent, time was devoted to check for possible anomalies in the data. There were two main concerning inconsistencies that were found which would influence the project.

One of the most concerning problems was a discrepancy in the number of distinct Brand IDs and Order IDs in the different tables (see Appendix C). Meaning that, when it came to joining the different tables on order to build the final dataset there would be a significant percentage of missing values. To solve this problem the model will be built by shortening the scope by disregarding 3 months of data that were the source for this problem.

The second incoherence was the presence of consecutive orders with the exact same details. In the '*OrderDetails'* table, there were orders with the same articles and quantities for each article made within seconds from each other by the same client. After consulting TIPS 4Y it was agreed a minimum time threshold of 1-minute difference between orders that had the exact same details, if it was below the agreed threshold, it should be considered as a glitch in the system and the data should be disregarded.

Once the data was cleaned and coherent, a new table was created which will be used to develop the recommendation model and data dashboard. This table consists of TIPS 4Y's orders and the respective products that were bought in each one with two levels of granularity: Articles (*article_ID*), which corresponded to a specific article; and Products, which corresponded to a parts family (*GenArtNr*). To create this table various tables were merged in a particular way (see Appendix D). To find the right item, the variables *Brand* and the *article_id* were concatenated into

a single key in order to get the right parts family (*GenartNr)* because it is possible that the same *article_id* could exist for different brands

Originally, the dataset used for the first take on the model was supposed to have the highest level of granularity, using articles. However, the volume of data was too great and extremely sparse, which raised a problem of lack of computational power, making this option unviable. Thus, it was agreed that lowering the granularity of the data was the best course of action hence, using the products to build the recommendation system. Nonetheless, the problems of volume and sparsity endured therefore, further steps had to be taken.

First, it was decided that it would be better to discard the products that were not bought that often and work with the 95% most bought products, reducing the number of products from 997 to 282.

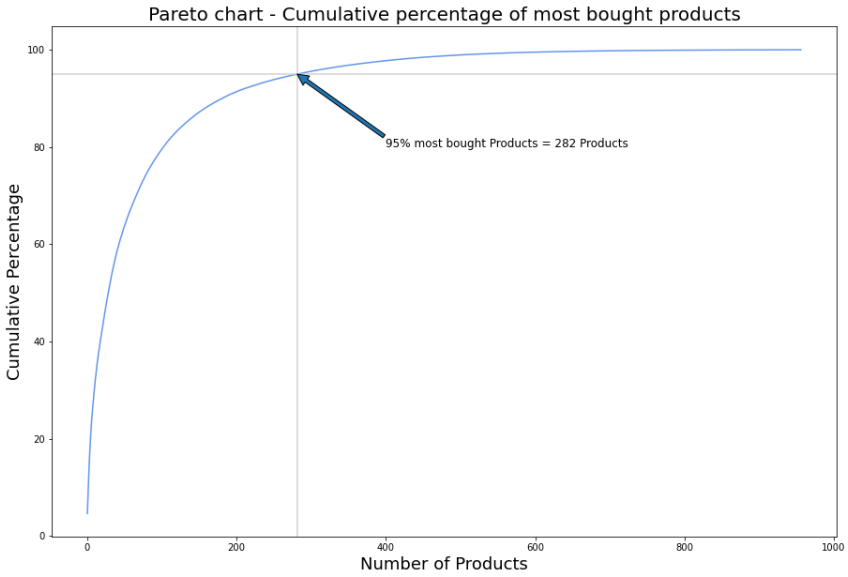

Figure 2: Pareto chart representing the Cumulative percentage of the most bought products

Secondly, orders that only had one product would not help find similarities between the products thus records would not add any value to the model and so, were also excluded. However,

after some discussion, it was also decided to disregard the orders with only two products in order to help reduce sparsity and due to lack of computational power. This being said, all the orders that had less than 3 different products were discarded and so the orders in the dataset decreased from 481,646 to 69,164.

3.2 Modeling

After the construction and preparation of the dataset, the next phase of the project could begin, *Modeling.* In figure 3, a framework is presented that shows the steps and flow of the steps that will be described in this phase.

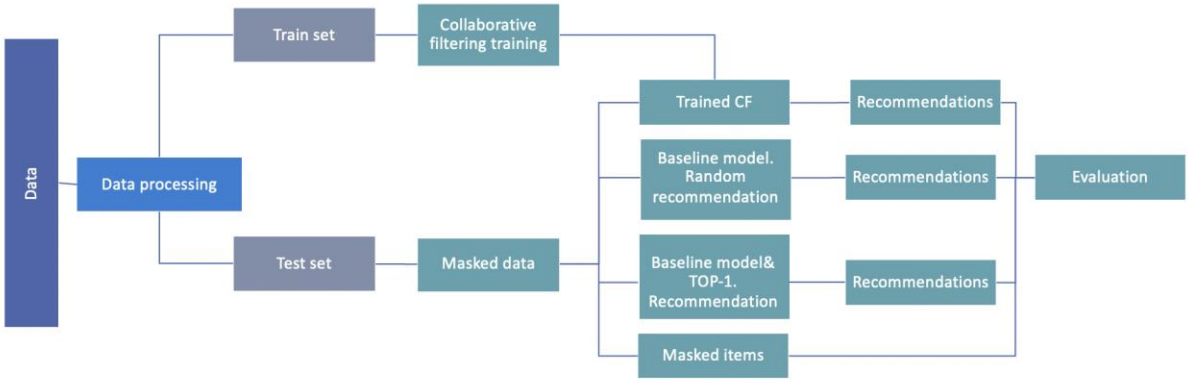

Figure 3 – Framework of the modeling Process

As a result of the literature review, the technique used to build the recommendation system was Memory-based Collaborative Filtering more specifically Item-based.

Given that our dataset did not have ratings, that are needed in a recommendation system, the appropriate approach was to consider the purchase of an item as a rating, meaning that the dataset would be transformed into binary matrix of orders vs products where:

$$Rating = \begin{cases} 1 = user\ bought\ the\ Product \\ 0 = user\ did\ not\ buy\ the\ \text{Product} \end{cases}$$

Thus, it's possible to consider the value 1 as a positive response, it was not possible though, to consider 0 as a negative response, given that this value only means that the user did not buy the product, not that it did not like it and so it was interpreted as a non-response. The focus needs to be on the positive responses.

The next step was data division where the dataset was divided into train and test sets with the common ratio of 70/30 respectively. In the training set, there was no manipulation of the data since it was used to train the model. However, the test set had to undergo some alterations, to later be used to evaluate how the model is performing. In each order of the test set one random product order was converted to zero, as if it had not been bought, and stored in a table called *'masked data'* where it would later be used to see if the model could recommend the item the exact product that was 'masked' from each order.

Next step was calculating the cosine similarity between products. This method measures the similarity between two vectors of an inner products space. It measures the cosine of the angle between two vectors and determines whether they are pointing in roughly the same direction. In other words, cosine similarity is a division between the dot product of vectors and the product of the Euclidean norms or magnitude of each vector (Richmond 2021) Then we optimized the algorithm to create a single product recommendation for each order.

After producing the recommendations, the model was ready for the evaluation process by comparing the recommendations to the masked products from the test set.

4. EVALUATION

Keeping in mind that this is a classification problem, the appropriate measures should be adopted to evaluate the models. To calculate the metrics used for classification problems, **Accuracy**, **Precision** and **Recall**, it is first needed to calculate the **Confusion Matrix** and interpret its values according to the context of this thesis:

- **True Positive** (TP) - where the recommendation made by the model is the same as the product that was masked.
- **True Negative** (TN) - all other products that were not recommended and were not the masked product.
- **False Negative** (FN) – the masked product does not equal the recommendation.
- **False Positive** (FP) **-** recommendation does not equal the Masked product

In this case, False positives and False Negatives are going to be the same, since if one recommendation is wrong then the recommendation made is going to be a false positive and the product that was supposed to be recommended is going to be a false negative.

In addition, given the highly imbalanced classes in our dataset, since a very small number of Products are bought per order, the number of zeros will be extremely high: 19,048,275 Nº of zeros versus 455,973 Nº of ones. Accuracy will be a poor metric to evaluate our model given that we will have high number of True negatives and we are interested in finding the True positives. Thus, the most appropriate metrics to deal with these cases are Precision and Recall. Putting these metrics into context, Precision tells us the proportion of the recommendations made that are actually correct whilst Recall will give us the ability of a model to find all the relevant cases within a dataset. However, given that False Positives and False Negatives have the same value, by looking at the formulas we can see that Recall and Precision will also have the same value. That being said for simplicity purposes it will only be mentioned a single metric, Precision.

$$\text{Precision} = \frac{True\ positive}{True\ postive + False\ positive}$$

Equation 1: Precision Formula

$$\text{Recall} = \frac{True\ positive}{True\ postive + False\ negative}$$

Equation 2: Recall Formula

It is important to mention another widely used metric in classification problems, especially when using Precision and Recall, the F1-Score. This metric is used to find a balance between Recall and Precision. Sasaki 2007, defined this metric as the harmonic mean of the models of Precision and Recall. In the context of this thesis, such metric is not of use. As previously mentioned, Recall and Precision present equal results, therefore, so will the F1 score.

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Equation 3: F1 Score Formula

Looking at the confusion matrixes for the Collaborative Filtering and the two baseline models, as predicted, the number of True Negatives is considerably higher than the other measures. Hence, if Accuracy had been chosen as a metric evaluation, all the models would seem to be the same, since all have more than 99% due to the True Negatives, clearly showing why Accuracy is not a good metric for imbalanced classes. Comparing the True Positives, we can see a big difference between the 3 models with Collaborative Filtering being far superior. Also leading to a lower False Negative and False Positive.

Random Product

	Predicted 0	Predicted 1
Actual 0	5,810,081	20,669
Actual 1	20,669	81

Table 1 – Confusion matrix for Baseline model Random Product

Most bought Product

	Predicted 0	Predicted 1
Actual 0	5,810,896	19,854
Actual 1	19,854	896

Table 2 – Confusion matrix for Baseline model most bought Product

Collaborative Filtering

	Predicted 0	Predicted 1
Actual 0	5,813,710	17,040
Actual 1	17,040	3,832

Table 3 – Confusion matrix for Baseline model Collaborative Filtering

Translating these values into the Precision metric the results clearly show that the Collaborative Filtering has a much higher Precision when compared to the other two baselines. Putting into perspective, Collaborative Filtering is 47.3 times more precise than recommending a random product and 4.3 times more than recommending the most bought product. That being said, this value is quite positive given that it is trying to predict the exact product that was masked from each order. Moreover, even though the product was wrongly predicted it might still be an extremely relevant recommendation, this will only be possible to measure once it is implemented.

	Random Product	Most Bought Product	Collaborative Filtering
Precision	0.39%	4.31%	18.46%

Table 4 – Results of Precision of the three models.

5. DIMENSIONALITY REDUCTION TECHNIQUES TO IMPROVE RECOMMENDATION SYSTEM

After the first results of the Collaborative Filtering, there may be room for improvement. Since the dataset has a large number of inputs the most appropriate step is to adopt Dimensionality Reduction techniques that might lead to better results. There are two main techniques used in Recommendation systems: **Principal Components Analysis (PCA)** and **Singular Value Decomposition (SVD)**.

Principal Components Analysis is a statistical technique that converts a group of observations from possibly correlated variables into a group of values named Principal Components (PCs), (Bokde 2015 et. al.). PCA can be considered as a technique that is used to generate non-correlated variables, that are a linear combination of the original values. (Jesus 2017). The process of choosing the number of PCs is dependent on how much information we are willing to lose. However, there are three criteria that are popularly used: **Kaiser's criteria**, where the PCs extracted need to have an eigenvalue greater than one; **Pearson's criteria**, where every PC is retained until 80% of the variance is explained; and **Scree Plots method** (Jesus 2017) which is similar to the elbow method when choosing the number of clusters

Singular Value Decomposition like the PCA decreases the dimensions of the data from *N* to *K* where the number of Ks cannot be higher than the original N Dimensions. It decomposes the data into three other matrices and extracts the factors from the factorization of a high-level matrix.

$$X = USV^T$$

Equation 4: Algebraic expression for SVD

Where *X* is the utility matrix, *U* is an orthogonal left singular matrix, that corresponds to the correlation between users or items, depending on the case, and the latent factors. *S* is a diagonal

matrix, that shows the strength of each latent factor (kumar 2021). Finally, *V* is a diagonal right singular matrix, which indicates the similarity between user or item, again depending on the case, and the latent factors. There are various variations of SVD, one that is commonly used is the Truncated SVD. The difference between the two techniques is that the n number of columns, from the factorization produced, can be indicated for a number of truncations whilst common SVD outputs the n columns of matrices.

In the context of this thesis and the way that the problem was tackled, it was imperative to choose the right method of Dimensionality Reduction given the extreme sparsity problem of the dataset. As previously mentioned, considering the purchases as ratings results in a big contrast between the zeros (19048275) and ones (455973), in fact, the ratio of ones in the whole dataset is 2.33%. This being said, the technique that better handles data sparsity should be adopted. Between the methods discussed, Singular Value Decomposition is the one that works better with sparse datasets (Chen 2020).

To perform Dimensionality Reduction, the first step is to decide which aspect of the data is the most appropriate to address, products or orders. Taking into consideration the goal of the model and its business application, appropriate and meaningful recommendations of products, applying Dimensionality Reduction to the 'items' would not make sense since the variable already represents the lowest data granularity available. Meaning the new dimensions extracted would be extremely difficult to explain and in terms of business application would not have any value since the purpose is to recommend the exact products. On the other hand, considering that the matrix being used is orders vs items and not the conventional user vs item, this problem does not exist since the relationship between orders is not relevant to the project.

5.1 Computing Singular Value Decomposition

To calculate the SVD, it was used the package *truncatedSVD* from the sklearn library. In order for the results to be comparable, it was used the same methodology and training set as in the Collaborative Filtering. Before applying this method, some data preparation had to take place. As previously explained, the dimension to be addressed was the orders, to do so, the dataset had to be transposed where the columns would be the orders and the rows would be the items.

The first attempt was to conduct the normal SVD to reduce the dimensions of the table. By setting the number of components at 282 (same as the number of products) when using the package *truncatesSVD*, is the same as performing the normal SVD since it's the maximum number of components possible.

The following steps were in line with the methodology previously used. With the resultant matrix, the cosine similarity matrix was calculated to then attempt to predict the items that were masked in the test set. The Recall of this method was 17,86%, which is lower than the Collaborative Filtering without the use of SVD. Meaning that Singular value decomposition does not improve the model, however, it doesn't particularly make it worse since the difference in the Recall/Precision between the two attempts is not that considerable.

The next attempt was to conduct the Truncated SVD so that experiments could be made with the different number of components used for the model and evaluate the results. To achieve the optimum number of components, an analysis must take place by resorting to Kaiser's and Pearson's criteria.

Figure 4 shows the cumulative percentage of the variance explained by the different number of components. By Pearson's criteria, at least the first 70 components should be extracted since they account for 80% of the variance of the data. However, the model should account for the most information possible, so the components to extract might be higher than this depending on the eigenvalues of the rest of the components.

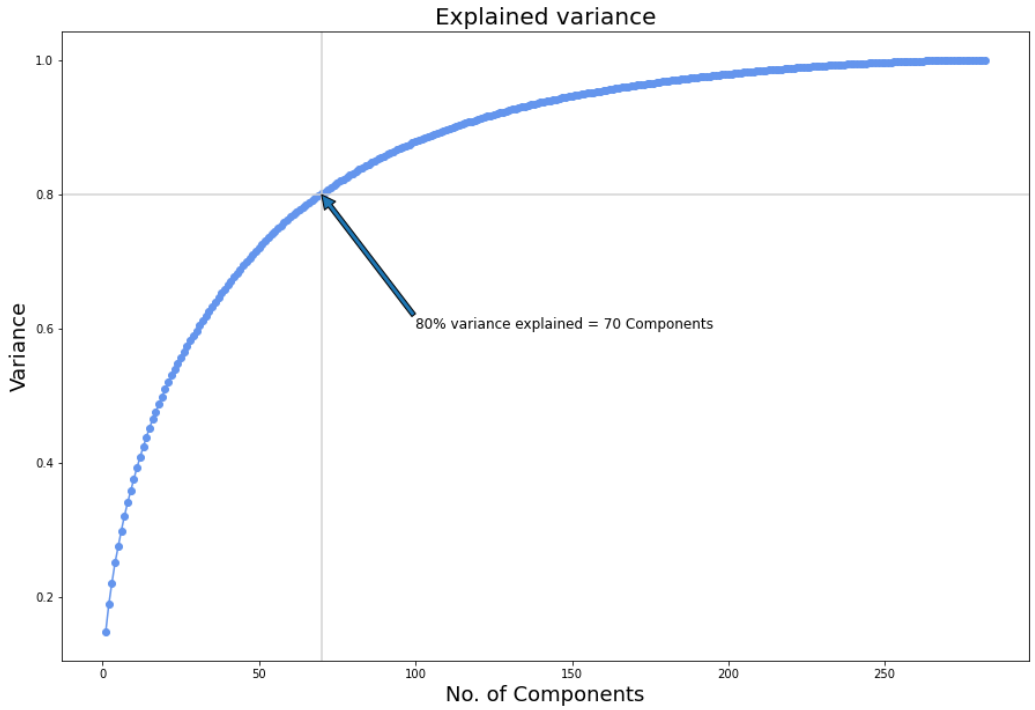

Figure 4 - Cumulative variance explained by each component

Following Kaiser's criteria, Figure 5 presents the eigenvalues for each component, the number of components to extract is 272. These results present a various number of components that, according to the criteria used, are relevant for the model.

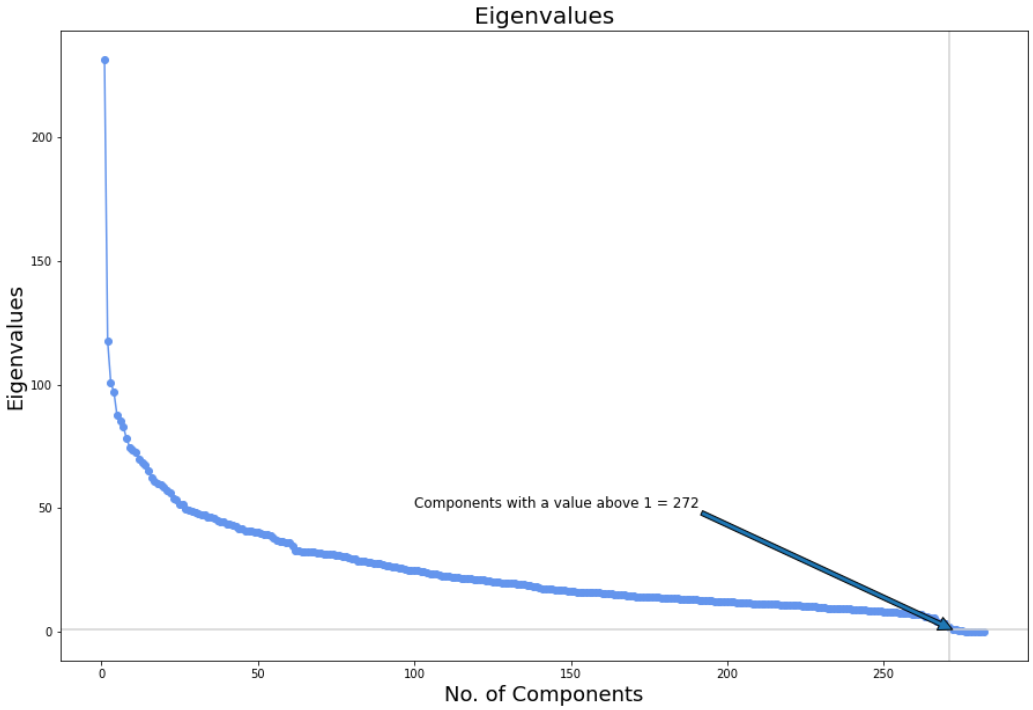

Figure 5 - Number of eigenvalues for each component

Given the results of the eigenvalues and the explained variance of the different components, it would be interesting to see how the Collaborative Filtering would respond using different numbers of relevant components. Following the same methodology, the results for using 70 components were substantially lower than the previous attempts with the Precision/Recall metrics accounting for 10.19%. Whilst the Truncated SVD with 272 components were exactly the same as the conventional SVD, with Precision/Recall at 17.86% however still lower than the original Collaborative Filtering.

Number of components	n =70	n = 272	collaborative filtering
Variance explained	80.05%	99.99%	-
Lowest Eigenvalues	31.29	1.29	-
Precision/Recall	10.19%	17.86%	18.46%

Table 5 – Results of the Collaborative filtering using truncated SVD with the different nº of components

5.2 Clustering

Clustering is a technique used for combining observation into groups (clusters) in way that is as homogeneous as possible by minimizing intra-cluster distances and each group should be different from other groups by maximizing inter-cluster distances with respect to a certain characteristic (Jesus 2017).

There are two main types of clustering algorithms, hierarchical and non-hierarchical (also known as Partitional clustering). One of the most used non-hierarchical algorithms is the **K-means**. This algorithm requires for the number of clusters to be set apriori which could be a problem since, usually, it is sensible to the initial seeds (Jesus 2017). To calculate the actual clusters k-means uses the Euclidian distance between each of the k centroids and each point in the data (Saggio 2016). However, K-means does not work properly with categorical data and, since the dataset being used is all binary, this technique would not be appropriate for this context. Therefore, a variation of the K-means that works with categorical data, called **K-modes**, must be used.

Although not a lot of information is available on k-modes, according to Huang 1998 it uses *"simple matching dissimilarity measure to deal with categorical objects, replaces the means of clusters with modes, and uses a frequency-based method to update modes in the clustering process to minimize the clustering cost function"*

To perform the K-modes clustering algorithm, it was used the package *kmodes* from the *kmodes* library. Following the same methodology as the SVD, after having our dataset prepped, the first step was to define the number of clusters we wanted to make, in order to reduce the number of orders. Usually, the elbow technique is applied to decide how many clusters should be used, however, K-modes takes a lot of time to run thus making the elbow technique not a viable solution. Therefore, a strategic number of clusters had to be set in order to only run the algorithm a few

times and get a sense optimum number of clusters. To evaluate the quality of the clusters it was used the silhouette score, where 1 would mean that the clusters are well constructed, 0 would mean the created clusters are not significant, and -1 where that clusters are assigned in the wrong way. Looking at table 6, we can see that the results for the different approaches are not favorable with all the scores being very close to 0 and some clusters even having a negative score. Concluding that this technique is not applicable to our dataset.

k	Silhouette score	Worst cluster	Best cluster
50	**0.0473**	**-0.3307**	0.5909
100	0.0451	-0.3395	0.5803
282	0.03143	-0.3758	**1.000**
1000	0.0141	-0.4781	**1.000**
5000	0.0199	-0.4756	**1.000**

Table 6 – silhouette scores for k-modes clustering algorithm

6. FUTURE WORKS

In the future, there are many different techniques and experiments that have been left out of this thesis due to a lack of time, computational power, and data. Due to the high dimensionality and volume of data, time was a scarce resource since some operations would require days to run. In fact, some would not even be possible to run, which occurred in various attempts throughout this thesis, due to being limited to the computational power of a personal computer.

One of the main approaches that should have a second look, with the appropriate resources, would be to consider using the actual articles instead of the parts family (GenartNr). In terms of business application, building the recommendation system with the articles would add substantially more value to the end-users than recommending a family of articles. In addition, having our dataset as a binary matrix created many problems and prevented the exploration of many techniques that do not handle binary data very well. Hence, a further examination should be made to consider an alternative approach by using orders and finding a workaround for not having actual ratings for the articles. Finding a methodology that would create a rating system based on the purchases that the client made.

Potential room for improvement of the recommendation system would concern attempting to construct a more complex model. Particularly, performing Neural Networks that have accomplished enormous success on Image Processing, speech recognition, and forecasting. Though it has been thoroughly analyzed its application on recommendation systems. A recent study made by He et. al. 2017, attempted an approach called Neural Collaborative filtering that "*replacing the inner product with a neural architecture that can learn an arbitrary function from data*" (He et. al. 2017, p. 1). An appealing aspect of their work that connects to the context of this thesis was its capability to perform matrix factorization. Their results showed that recommendation

performance improved by using deeper layers in the neural networks. That being said, it would be interesting to study this approach, however, Neural networks are very complex techniques and perhaps TIPS 4Y does not yet have the need for such a complex model and a simpler approach is more appropriate.

Finally, the most important aspect that was not possible to accomplish was the live evaluation of the recommendation system after it was implemented in order to actually understand if the model was performing according to expectations and if in fact this thesis was a success and solved TIPS 4Y's problem. Or on the contrary, was underperforming and the model had to be rebuilt using another approach.

7. CONCLUSION

In this thesis, the problem presented by TIPS 4Y was that they were losing possible revenue due to due to lack of information on its customers purchasing habits and a tool to benefit from that information. The problem was addressed by attempting to build a recommendation system. After careful analysis of the data available and thorough research on the various techniques available on the matter, it was evident that the most appropriate approach was to construct an item-based Collaborative filtering. However, some adjustments had to be made since there were no ratings for the articles, thus a binary matrix was developed between the articles and the orders made by TIPS 4Y's clients. Also, due to the high volume and dimensionality of the data, the granularity of the data had to be changed due to limited computational power. The initial results for the developed recommendation system were far better than initially anticipated, especially compared to the two baseline models that were created for the purpose of comparison: the Collaborative Filtering was 47.3 times more precise than recommending a random product and 4.3 times more than recommending the most bought product. Attempts to improve the model through Dimensionality Reduction were made by resorting to Singular Value Decomposition and the variant Truncated Singular Value Decomposition. However, the results of the various attempts were not promising as they did not perform better than the original model. Nonetheless, the simple Collaborative Filtering developed showed great results and will most likely have an impact on the TIPS 4Y's revenue.

8. REFERENCES

Nadali, Ahmad, Elham Naghizadeh Kakhky, and Hamid Eslami Nosratabadi. "Evaluating the Success Level of Data Mining Projects Based on CRISP-DM Methodology by a Fuzzy Expert System." 2011 3rd International Conference on Electronics Computer Technology, 2011. doi:10.1109/icectech.2011.5942073.

Isinkaye, F.o., Y.o. Folajimi, and B.a. Ojokoh. "Recommendation Systems: Principles, Methods and Evaluation." Egyptian Informatics Journal 16, no. 3 (2015): 261-73. doi:10.1016/j.eij.2015.06.005.

Linden, Greg, Brent Smith, and Jeremy York. "Amazon.com Recommendations: Item-to-item Collaborative Filtering." IEEE Xplore. Jan. & Feb. 2003. Accessed December 16, 2021. https://ieeexplore.ieee.org/abstract/document/1167344/.

Vatsal. "Recommendation Systems Explained." Medium. November 03, 2021. Accessed November 15, 2021. https://towardsdatascience.com/recommendation-systems-explained-a42fc60591ed.

Alake, Richmond. "Understanding Cosine Similarity And Its Application." Medium. November 03, 2021. Accessed December 16, 2021. https://towardsdatascience.com/understanding-cosine-similarity-and-its-application-fd42f585296a.

Sasaki, Yutaka. "The Truth of the F-measure." School of Computer Science, University of Manchester, October 26, 2007, 1-5. Accessed December 8, 2021. https://www.cs.odu.edu/~mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf.

Bokde, Dheeraj Kumar, Sheetal Girase, and Debajyoti Mukhopadhyay. "An Item-Based Collaborative Filtering Using Dimensionality Reduction Techniques on Mahout Framework." ResearchGate, March 2015, 1-7. https://www.researchgate.net/publication/274012078_An_Item-Based_Collaborative_Filtering_using_Dimensionality_Reduction_Techniques_on_Mahout_Framework.

Dr. Vaibhav Kumar Vaibhav Kumar Has Experience in the Field of Data Science and Machine Learning, and Dr.Vaibhav Kumar. "Singular Value Decomposition (SVD) & Its Application In Recommender System." Analytics India Magazine. January 12, 2021. Accessed December 15, 2021. https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system/.

Jesus, Frederico Cruz. "Data Analysis." Data Analysis Course, Nova Information Management School, Lisbon, 2017.

Chen, Denise. "Recommender System-singular Value Decomposition (SVD) & Truncated SVD." Medium. August 06, 2020. Accessed December 15, 2021. https://towardsdatascience.com/recommender-system-singular-value-decomposition-svd-truncated-svd-97096338f361.

Dunteman, George Henry. Principal Components Analysis. 69th ed. Newbury Park, CA: Sage, 1989.

Alessia, Saggio. "Into the World of Clustering Algorithms: K-means, K-modes and K-prototypes." AMVA4NewPhysics. October 26, 2016. Accessed December 10, 2021. https://amva4newphysics.wordpress.com/2016/10/26/into-the-world-of-clustering-algorithms-k-means-k-modes-and-k-prototypes/comment-page-1/.

He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. "Neural Collaborative Filtering." Proceedings of the 26th International Conference on World Wide Web, 2017. doi:10.1145/3038912.3052569.

9. APPENDIXES

8.1 Appendix A – Data Dictionary (Only available online or contact authors)

Data Dictionary - Tips4y.xlsx

8.2 Appendix B – Database

Tables	Variables	Description	type	Number of distinct values	Missing values
Article	Id	Unique ID for each article	Categorical	119 401	0%
	brand_id	Unique ID for each brand	Categorical	181	0%
	article_id	business ID for article	Categorical	116 808	0%
Brand table1	BrandID	Unique ID for each brand	Categorical	68	0%
	Description	Brand Name	Categorical	68	0%
GenArt table	GenArtNr	Parts family Unique ID	Categorical	6 814	0%
	Description	Parts Family Description	Categorical	5 979	0%
Order Details	order_id	Unique ID for each order	Categorical	455 237	0%
	article_id	Unique ID for each article	Categorical	77 636	0%
	quantity	Part Quantity	Quantitative	45	0%
	price_unit	Part Unit Price	Quantitative	13 735	0%
Orders1	id	Unique ID for each order	Categorical	590 318	0%
	supplier_id	Supplier ID	Categorical	3	0%
	client_id	Client ID	Categorical	723	0%
	orderdate	Date when the order was made	Quantitative	565 819	0%
	ordertotal	Total Order Value	Quantitative	32 777	0%
TD article	article_id	business ID for article	Categorical	106 2230	0%
	BrandID	Unique ID for each brand	Categorical	75	0%
	GenArtNr	Parts Family Unique ID	Categorical	2 794	0%

8.3 Appendix C – Discrepancy in the number of Brand ID's and Order ID's

Table/ Number of distinct	Brand IDs	Order IDs
Article table	181	-
Brandtable1	68	-
TD_article	75	-
Orders1	-	590 318
OrderDetails	-	455 237

8.4 Appendix D – Merged tables keys used are color-coded