ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Bachelor Thesis International Econometrics and Operations Research

Using purchase histories for predicting the next market basket

Octavian Maradin

472761

Supervisor: Jiawei Fu Second assessor: Anoek Castelein

Date final version: July 2020

Abstract

Predicting the next market basket accurately present various benefits for both brick and mortar shops, as well as for online stores. They include advantages for both the marketing and logistics of such a company. This research focuses on searching whether the next market basket bought by a customer could be predicted using his personal purchase history, as well as the acquisition histories of other customers. Several concepts are introduced for estimating the similarity between different market baskets and assessing the closeness of various purchase histories. The methods are tested on samples of the two versions of the Instacart dataset and on an entire e-commerce dataset. It is found that predicting the next market basket is more facile by using a baseline method when limited data is available or computational power is limited.

Contents

1 Introduction

Suggesting products to customers in a personalized manner is receiving more attention in the marketing field. An efficient implementation of this task can offer companies a competitive advantage. According to a study, 80% of the respondents claimed that they stopped doing business with a specific company because of a deficient customer experience (Redbord, 2019). Large retail shops could offer a better client experience by recommending relevant products to the individual and, thus, gain a strategic advantage when compared to other shops. Furthermore, understanding the demand of the clients could lead to better business solutions. Predicting the next market basket is one of the research areas which studies methods to accurately suggest the next set of items a person buys (Guidotti, Rossetti, Pappalardo, Giannotti, & Pedreschi, 2017).

Forecasting the next market basket has an application in the sector of e-commerce as well. Individuals tend to choose this method of buying more often, as this sector experienced a tremendous growth in the last 30 years. According to Forbes, e-commerce represented 14.3% of the total retail sales in 2018 (Schroeder, 2018). Thus. finding ways to improve the customer service of this business represents an important topic. The application of predicting the next market basket can be used in direct marketing campaigns. In this way, companies could attract more clients by offering better products recommendations.

This research aims to find a fast and precise method to predict the next market basket. In this sense, the research questions is formulated as follows:

How can we predict the next market basket accurately and efficiently using the global and personal purchase histories?

The focus will be put in replicating and consolidating the innovative study of Kraus and Feuerriegel (2019), which brings a new method of approximating next market basket of a customer with state-of-the-art results. The method in this study use purchase histories across all customers to predict the next market basket, as well as the personal acquisition history of the client. It is a combination of using global and personal features of purchasing behaviors. Kraus and Feuerriegel (2019) report a significant improvement in the prediction scores compared to the other methods proposed in the literature. An improvement of 4.0

times is reported compared to the scores of baseline methods

The report begins by presenting the literature. Then, the data is presented. This research first tests the methods on the proposed dataset of Kraus and Feuerriegel(2019). They include two versions of Instacart dataset[1]. As an extension, another e-commerce purchases dataset will be considered released by London South Bank University[2]. This dataset is considered as a result of the growing importance of the e-commerce industry as explained.

In the methods section, all the steps required for this method will be shown. The method used in this paper replicate the steps described in the original paper. The first step is creating vector representations of the products within the market basket. Then, it is explained how to compute the difference between two market baskets represented in a vector space (Tran, Vo, Phung, & Vo, 2016). Subsequently, as the data in this context is ordered in time, a method, generally used for computing the difference between the two time series sequences, will be introduced for assessing the difference between different purchase histories. The prediction of the next market basket will be given by the most similar purchase histories in relation to the personal acquisition history of the customer.

In the results section, the outcome of this method will be discussed. It is found that for smaller samples of the data, the algorithm does not outperform the results obtained by the baseline methods. The same conclusion is drawn for all the datasets considered in this research. However, the results obtained by considering only a sample of the two Instacart versions are comparable to the results reported by Kraus and Feuerriegel(2019) on the whole dataset. A general conclusion is drawn that the baseline methods are a better solution in the case of limited computational power or smaller datasets. In the end, several limitations of this method, as well as a general analysis will be included in the conclusion section. It will be further discussed how this method can be improved.

2 Literature

In order to assess the performance of the methods, it is usual for the proposed methods to be compared to some baseline approaches. Gupta and Mamtora (2014) present some

[1]https://www.kaggle.com/c/instacart-market-basket-analysis/data

[2]https://www.kaggle.com/carrie1/ecommerce-data

of these methods. For example, *same as last trip* method simply gives the prediction that an individual will buy exactly the same products as his last shopping list. Another method considered is *TOP*. This method returns the most popular products that the client has bought in all his acquisitions and its accuracy is limited. A more sophisticated baseline procedure uses the so-called association rules. In this approach, the probability that a product is bought is computed using the last products bought by the client. It is based on support, which measures how many times the products have been bought together and confidence, which asserts the accuracy of the estimation. However, this method also has limited results.

One area of research in the market basket analysis focused on the sequential aspect of purchasing products. One of the most known methods is to use Markov Chains (MC), which uses only the purchase history of the customer (Zimdars, Chickering, & Meek, 2013). This method estimates the next state in the chain by using a decision-tree approximation. Another branch of research is based on collaborative filtering. Collaborative filtering is a branch of the recommender systems, usually used in contexts where customers rate the products and the suggestions are estimated using this extra information. The adapted version for market basket analysis uses non-matrix factorization (*NMF*) over items and all the clients (Lee & Seung, 2001).

Then, a part of the literature suggests combining non-matrix factorization with Markov Chains (*FPMC*) such that both the sequential feature of the purchases and the tastes of the customers are captured in the model (Rendle, Freudenthaler, & Schmidt-Thieme, 2010). Furthermore, in this hybrid model, the factorization is created for each client, meaning that a specific transition matrix is used for each individual. Hierarchical representation model (*HRM*) is an approach which also tries to captures both the personal sequential purchases, as well as the general preferences of the individuals (Wang et al., 2015). It's novel idea is to describe the users and the items bought in a vector. It employs a two-layer structure to characterize users and items over the previous acquisition of a client to predict the next market basket. The main issue with HRM is that it does not include in the model the information across baskets. Thus, in order to solve for this problem, a method based on recurrent neural networks was proposed which captures both the personal purchase history and the global sequential features in the dataset (Yu, Liu, Wu, Wang, & Tan, 2016). Neural networks are a machine learning classification algorithm based on the structure of the human brain. Recurrent neural networks allows observations to be included in more than one category which makes it suitable for shopping lists, as one product will be part of many baskets.

A recent approach which takes into account only the data of the customer is Temporal Annotated Recurring Sequences Basket Predictor (*TBP*) (Guidotti, Rossetti, Pappalardo, Giannotti, & Pedreschi, 2018). It includes different elements of the purchasing habits of the customer that have an impact on the decision process such as recurrency.

The work of Kraus and Feuerriegel(2019) continue the series of studies which are based on the sequential aspect of the purchases. It first computes the similarity of the market baskets across the customers. Then, the purchase history similarity of the customers is computed to predict the next market basket to achieve a new state-of-the-art result.

All the studies are generally aimed for large datasets. However, in some cases, the amount of data available is not always substantial. Thus, this research aims to check the efficiency and accuracy of the results of Kraus and Feuerriegel(2019) in smaller contexts.

3 Data

This research will be first tested on two versions of Instacart dataset, in the same manner as Kraus and Feuerriegel (2019). The two versions are named Instacart product and Instacart aisles. Subsequently, as an extension to the original paper, the method will also be tested on an e-commerce dataset.

The Instacart dataset included food products and it is often used in the field of predicting the next market basket. The dataset contains 3,412,0841 orders with approximately 50,000 products which are classified in 134 aisles (categories). In Figure 1, the number of orders for per user can be observed. It can be seen that most of the individuals buy below 10 market baskets. However, in this research only the customer over that threshold for this dataset in a similar way as the original paper, which results in less than 20% of the whole dataset to be used.

On a similar note to the original paper, the baskets retrieved from the Instacart dataset will be selected. As already mentioned, for both versions of the dataset, the customers

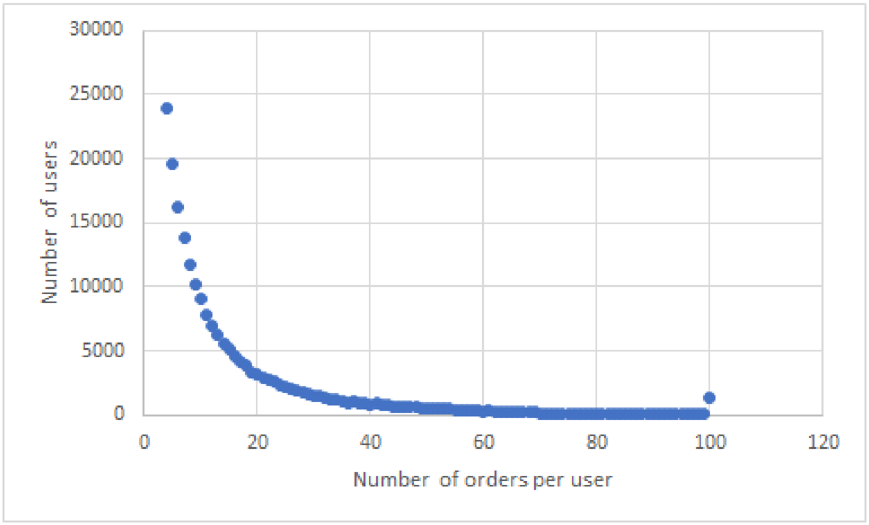

Figure 1: Total number of orders per customer for Instacart dataset

considered are only the ones who have purchased at least ten market baskets. Furthermore, each market basket need to contain at least 5 products.

Instacart product considers the orders at product level, referring to the fact that the name of the product acquired will be predicted. For example, the product named 'Chocolate Sandwich Cookies' is categorised in the aisle 'cookies cakes'. Thus, for this version of the dataset, the market basket will include the product name.

In the case of product level predictions, another filter than the ones mentioned previously will be considered. Specifically, only the items which are in the list of the 500 most frequently purchased products are considered in this research. After applying the filters and constructing the market baskets, 26,001 customers and 609851 market baskets have resulted. Out of this dataset, a sample of 1200 purchase histories is selected. This sample is split in 1000 training observations, 100 validation data points and the rest of 100 purchase histories have been included in the test set.

Instacart aisles ignores the actual name of the product bought and builds the baskets by including the aisle of the product. Thus, in this case, the method will aim to predict the next category of products bought. Considering the example earlier, the basket would include 'cookies cakes' for the product 'Chocolate Sandwich Cookies'. After applying the two filters described earlier, 67145 customers and 966496 orders are extracted. Then, a sample of 1200 acquisition histories is considered, which is split in three sets of 1000, 100 and 100 customers corresponding to the training, validation and test sets.

As an extension, an **E-Commerce** dataset will be considered. It was published by the London South Bank University and it included actual transactions in the period 2010 to 2011. Moreover, the data is registered from shops which sell their goods only through online means. It differs from Instacart, as the products in this dataset are various type of gifts rather than food items.

For this dataset, only the 500 most popular products are considered. Furthermore, as the dataset is significantly smaller than the Instacart one, all the market baskets with more than 4 products will be taken into account. Moreover, each client should have at least 3 baskets acquired. After applying these filters, 1062 purchase histories have resulted. Out of these total number of customers, 885 are used for training, 89 are included in the validation set and 88 will be used for the test set.

4 Methodology

The problem of estimating the next **market basket** is formally formulated as follows. Let v goods be sold by a company, listed in a set $I = \{i_1, \dots, i_v\}$. A given customer c purchases items from this store z times, given in the purchase history $B_c = \{{b_c}^1, \dots, {b_c}^z\}$, with ${b_c}^i \subseteq I$ and ${b_c}^1, \dots, {b_c}^z$ ordered in time. As there are n customers, the set of clients will be represented by $B = \{B_1, B_2, \dots, B_n\}$, where B_i represents the purchasing history of client i. The task which follows is to predict the next market basket of the client c namely ${b_c}^{z+1}$ correctly, with ${b_c}^{z+1} \subseteq I$. This implies naming the whole set of goods that the person will buy.

The algorithm starts with mapping the products within a market basket onto a vector space. This step ensures returning a vector for each product which ensures that products which are similar to each other are mapped adjacently. Then, the distance between two market baskets is computed using these vector representations by making use of the **Wasserstein distance**. In order to fasten the computations, a lower bound is proposed in this step. Calculating the difference between purchase histories follows, computed with **subsequential dynamic time warping**. Rather than a simple prediction of the next market basket based on the distances computed, a more elaborate prediction is made with the **k-neighbors algorithm**. All the steps have been implemented in Python 3.6.

4.1 Item embeddings

The idea of mapping words into vector spaces has been introduced in the field of natural language processing in order to group similar words together (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). One of the most used models lately for representing words as vectors is the Skip-gram model. In order to understand this algorithm, the concept of artificial neural networks has to be understood. For an introduction in this algorithm, check Appendix A.

Skip-gram model

Given a sentence or a document, the aim of this model is to use the formed vectors in order to predict adjacent words. In the given context, this translates to finding similar products within the same market basket.

The algorithm uses a more complex architecture of neural networks, which prove to be efficient and accurate estimations in this context (Mikolov, Chen, Corrado, & Dean, 2013). It is inspired from the **Feedforward neural net language model (NNLM)** which estimate the statistical model of the distribution of word sequences (Bengio, Ducharme, Vincent, & Jauvin, 2003). In other words, NNLM algorithm estimates the probability of a certain word belonging to a context. It differs from the usual neural network by introducing a projection layer. Thus, the input layer retains all the words in a sequence of words, which are then mapped to some initial vector space. This initialization is made with vectors of free parameters. Following this step, the neural networks process the projected vectors and return the word sequences distribution.

Skip-gram differs from the described algorithm mainly through its different objective. Instead of aiming to predict a word based on the other terms around it, Skip-gram uses a word to predict the possible words adjacent to it. It seeks the words which have the highest probability to be in a certain range in the adjacency of the current word.

Mathematically, the aim of the skip-gram model translates to mapping words onto vector spaces such that the prediction of the surrounding words is accurate (Mikolov, Sutskever, et

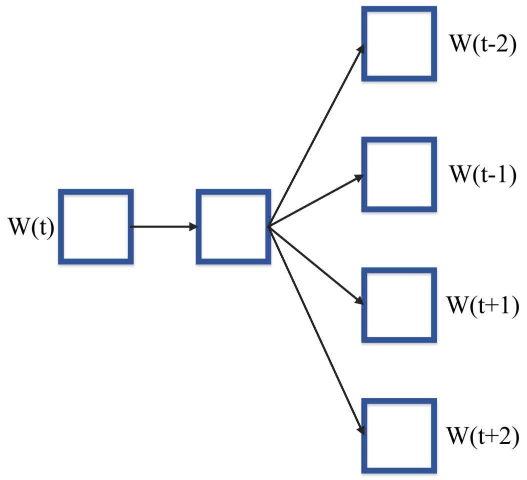

Figure 2: Skip-gram architecture

al., 2013). Therefore, the objective function is defined as follows. Thus, in the context of predicting the next market basket, the algorithm works as follows. Having as input a basket of items $b = \{i_1, i_2, \ldots, i_c\}$ the function maximised is

$$\frac{1}{M}\sum_{i=1}^{M}\sum_{-c\leq l\leq c, l\neq 0} \log p(w_{i+l} \mid w_i), \tag{1}$$

where c is the dimension of the basket. Furthermore, the function $p(w_{i+l} \mid w_i)$ takes the following form in this context:

$$p(w_l \mid w_z) = \frac{exp(X_{w_l}^{I} Y_{w_z})}{\sum_{w=1}^{T} exp(X_{w}^{T} Y_{w_z})}, \tag{2}$$

where X_{wt} and Y_{wt} are the projected and output vectors corresponding to the item i_t and I is the number of all the items considered. As the theoretical formulation of this model is highly computational expensive, a number of approximation methods have been introduced. In this research, **negative sampling** (NEG) has been used.

NEG represents a simplified version of the Noise Contrastive Estimation algorithm, which manages to find the noise in the data with the use of logistic regression (Gutmann & Hyvärinen, 2012). The simplification made does not affect the quality of the vectors obtained with Skip-gram. Formally, an approximation is made for computing $\log p(w_l \mid w_z)$, such that it is equal to:

$$\log \sigma(X_{w_l}^T Y_{w_z}) + \sum_{j=1}^{K} \mathbb{E}_{w_j \sim P_n(w)} \cdot \log \sigma(-Y_{w_z}^T X_{w_l}),$$

where $P_n(w)$ represents the noise distribution, defined through the logistic distribution and K is a number used when simulating the data.

The algorithm returns a series of vectors such that similar products are mapped closer together in term of the cosine similarity. The cosine similarity between two vectors results by computing the angle between them and applying the cosine function on this angle. Thus, it could return an intuitive result close to 1.0 if the vectors are close to each other or 0.0 if the two vectors are orthogonal (Singhal et al., 2001).

4.2 Wasserstein distance

After mapping each product onto a vector space, computing the Wasserstein distance between market baskets follows. Given the mappings of the products of two market basket on the vector space S, $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$, where X and Y represent two market baskets and $x_i, , x_n$ and $y_1, , y_m$ are the items contained in them, the Wasserstein distance of order p is defined as (Tran et al., 2016):

$$d_W^{(p)} \overset{def}{=} \min_C \left(\sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} d(x_i, y_j)^p \right)^{\frac{1}{p}} \tag{3}$$

where C is a transportation matrix and d is the euclidean distance in this context. Furthermore, the elements of C satisfy the following conditions:

$$c_{i,j} \geq 0 \text{ for each } 1 \leq i \leq n, 1 \leq j \leq m \tag{4}$$

$$\sum_{i=1}^{n} c_{i,j} = \frac{1}{m}, \text{ for all j=1,\ldots, m} \tag{5}$$

$$\sum_{j=1}^{m} c_{i,j} = \frac{1}{n} \text{ for all i=1,\ldots, n} \tag{6}$$

It is important to note that when any of the sets is empty, $d_W^{(p)}$ will be ∞. However, in this research, the Wasserstein distance is computed only for baskets with more than 5 products which means this exception is not applied.

Computing the Wasserstein distance can be computationally expensive, especially that in this research, millions of market baskets will be included. The best average time complexity for computing this distance between two word sequences is $O(p^3 \log p)$, where p is defined as the total number of unique words in the two word sequences (Kusner, Sun, Kolkin, & Weinberger, 2015). Therefore, in order to make this method practical, it is required to fasten the operations. In this sense, a lower bound of this distance will be considered which will make it possible to prune away a majority of possible similar acquisition histories.

Following Kusner et al. (2015), a lower bound is obtained by adapting their results for this context. In this way, the following two lower bounds result, denoted LB_1 and LB_2. Given two market baskets $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$, it holds that

$$d_W^{(p)}(X,Y) \geq \sum_{j=1}^{m} \min_{k=1,\ldots,n} d(x_k, y_j)^p \frac{1}{m} = LB_1 \tag{7}$$

$$d_W^{(p)}(X,Y) \geq \sum_{i=1}^{n} \min_{k=1,\ldots,m} d(x_i, y_k)^p \frac{1}{n} = LB_2 \tag{8}$$

The proof of the first lower bound follows. The second lower bound is proved in the same manner and thus, the proof is skipped. A relaxed form of the optimization problem of the Wasserstein distance can be obtained by removing (6). Furthermore, it is known that by removing one or more restrictions in a constrained optimization problem, the solution to the relaxed problem will give a lower bound to the initial one. The resulting problem is:

$$d_W'^{(p)} \overset{def}{=} \min_{C'} \left(\sum_{i=1}^{n} \sum_{j=1}^{m} c'_{i,j} d(x_i, y_j)^p \right)^{\frac{1}{p}} \tag{9}$$

where $c_{i,j}$ are elements of C and subject to the following conditions:

$$c'_{i,j} \geq 0 \text{ for each } 1 \leq i \leq n, 1 \leq j \leq m \tag{10}$$

$$\sum_{i=1}^{n} c'_{i,j} = \frac{1}{m}, \text{ for all j=1,..., m} \tag{11}$$

First, it can be observed that the solution of $(d_W'^{(p)})^p$ is the same as the one for $d_W'^{(p)}$. The solution of the optimization problem $(d_W'^{(p)})^p$ can be obtained in the following way. Considering C' a feasible matrix for (9), then:

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{m} c'_{i,j} d(x_i, y_j)^p &\geq \sum_{i=1}^{n}\sum_{j=1}^{m} \min_{k=1,\ldots,n} c'_{i,j} d(x_k, y_j)^p \\ &= \sum_{i=1}^{n} c'_{i,j} \sum_{j=1}^{m} \min_{k=1,\ldots,n} d(x_k, y_j)^p \\ &= \frac{1}{m}\sum_{j=1}^{m} \min_{k=1,\ldots,n} d(x_k, y_j)^p\end{aligned}$$

The obtained lower bounds are used for selecting promising market baskets of other purchase histories. The lower bound is first computed and only in the case when it is lower than the nearest candidates, then the exact Wasserstein distance is computed. Rather than choosing one of the lower bounds described, a stricter approach will be considered by taking the maximum of the two lower bounds ($LB = max\{LB_1, LB_2\}$). Using this relaxation method, the time complexity of this step is reduced to $O(p^2)$, compared to the $O(p^3 \log p)$ obtained when computing the exact Wasserstein distance.

4.3 Subsequence dynamic time warping

After previously computing the distances between different market baskets, it is now necessary to identify the closest sequence of acquisitions. In this sense, the dynamic time warping algorithm will be used for checking similarity between two temporal sequences. First, dynamic time warping will be introduced in its original form. Then, the adapted form of the algorithm, subsequential dynamic time warping will be explained. Furthermore, an efficient method of implementation known as star-padding technique will be used in this research (Sakurai, Faloutsos, & Yamamuro, 2007).

The main advantage of this algorithm is that it allows scaling along the axis. This means that it can capture similarity even when individuals change their purchasing habits at a different pace. This fits the current context, as there exists individuals who repeat their purchases many times.

Dynamic time warping (DTW)

Given two purchase sequences of customer X and Y, $B_X = \{b_X^1, \ldots, b_X^n\}$ and $B_Y = \{b_Y^1, \ldots, b_Y^m\}$ of length n and m, the DTW distance $D(X, Y)$ is defined as:

$$D(X, Y) = f(n, m) \tag{12}$$

where f is computed in the following way, with $i = 1, \ldots, n$ and $j = 1, \ldots, m$:

$$f(i, j) = d_W^{(p)}(b_X^i, b_Y^j) + min(f(i-1, j), f(i, j-1), f(i-1, j-1)) \tag{13}$$

$$f(0, 0) = 0, f(i, 0) = f(0, j) = \infty \tag{14}$$

The time complexity of DTW is $O(nm)$, since it represents the dimensions of the matrix. An advantage of this method is that it can capture the non-linear relation between the sequences considered. However, a main limitation of this approach is that it requires the first market baskets and the last ones of the two customers to be relatively close. In this context, this feature limits the results as there exists big differences in the number of purchases done by customers.

Subsequential dinamyc time warping (SDTW)

In order to solve the issue mentioned and implement a more flexible algorithm, SDTW will be considered. The main difference compared to the DTW is that now, subsequences of the longer purchase history will be used. Specifically, for $n > m$, DTW would be applied for the subsequence $B_X(l : z)$, with $1 \leq l \leq z \leq n$, and B_Y. The goal is to find the most similar purchase subsequence.

A naive implementation of this method would be to create subsequences with each element of B_X as the starting point and then applying DTW for each subsequence and B_Y. However, the time complexity achieved is impractical at $O(n^2m)$. Thus, star-padding technique introduced by Sakurai et al. (2007) will be used instead which achieves a complexity of $O(nm)$. The main point of this method is to insert a special value, that always returns zero distance. Thus, this translates in adding to the sequence B_X, a market basket b_X^0 with its vector representation made out of the intervals $(-\infty : +\infty)$. In this way, any distance from another market basket towards b_X^0 will be 0. In the end, the DTW algorithm will be applied to B_Y and the newly formed B'_X:

$$B'_X = \{b_X^0, b_X^1, \ldots, b_X^n\}$$
$$b_X^0 = (-\infty : +\infty)$$

As B'_X will be used, the matrix D will be computed in the following manner:

$$D(i, j) = f(i, j) \tag{15}$$

where f is computed in the following way, with $i = 1, \ldots, n$ and $j = 1, \ldots, m$:

$$f(i, j) = d_W^{(p)}(b_X^i, b_Y^j) + min(f(i-1, j), f(i, j-1), f(i-1, j-1)) \tag{16}$$

$$f(0, 0) = f(i, 0) = 0, f(0, j) = \infty \tag{17}$$

The final distance will result by taking the minimum of $D(i, m)$ with $i = 1, \ldots, n$.

4.4 Predicting with k-nearest neighbor

Instead of simply returning the basket in regards to the most similar purchase history, a different approach will be taken. In this sense, an adapted form of the **k-nearest neighbours** (k-NN) algorithm will be considered.

The method of k-NN usually works as follows. First, the most similar k training examples in regards to the test value are computed. Then, the decision of inferring the unknown variable is made by taking the majority class of the k examples.

In this context, the algorithm will be adapted accordingly. For a customer c, the k most similar subsequence purchase histories will be considered as returned by the SDTW. Formally, the k subsequences will be included in the set $N = \{B_1(i_x^1 : i_y^1), \ldots, B_k(i_x^k : i_y^k)\}$ with $1 \leq i_x^j \leq i_y^j$ and $i = 1, \ldots, k$ and i_x^k and i_y^k representing the first and last market baskets, respectively, of the purchasing subsequence of customer k.

The hyperparameter k will be tuned on the set of values $\{1, 2, 5, 10, 20\}$. In the case of 1-nearest neighbor search, the predicted basket will be $B_j[i_y^j + 1]$, where B_j represents the most similar purchase subsequence. For the other values of k, the most common goods from the considered subsequences will be included in the predicted basket. Moreover, in this case, the number of items in the predicted basket for customer c is equal to the average size of the market baskets previously bought by c.

Another feature will be considered when predicting the next market basket, named as the **Similarity-Aware Prediction**(Kraus & Feuerriegel, 2019). This implies computing the average SDTW distance $d(B_X, B'_Y)$ towards the potential k nearest neighbors, where B_X represents the purchase history of customer c and B'_Y refers to the purchase histories of the potential nearest-neighbors. If this distance is over a threshold τ, then the prediction of the next market basket will be made through personal top items baseline which is defined below. Moreover, τ represents a hyperparameter and its tunning will be detailed in the results section. The reason of implementing fallback prediction is that some customers have a specific purchasing behavior, which cannot be observed through the method proposed.

4.5 Baselines

In order to understand better the performance of the method proposed, it will be compared to a series of baseline methods universally used in the literature (Guidotti et al., 2018).

Global top items: This method creates a list of the most frequent bought items among all the customers. Then, it returns first n_c products of this list for a customer c, where n_c depends on each client.

Personal top items: Instead of looking at the most frequent items bought for all customer, this algorithm creates a list of most popular products of customer c. Then, it returns the first n_c products from it.

Repurchase last basket: This method will simply assume that the client will buy exactly the same items as the previous purchase. Specifically, for a purchase history $B_c = \{b_c^1, \ldots, b_c^n\}$, the predicted market basket ${b'}_c^{n+1}$ will be equal to b_c^n.

4.6 Measuring the accuracy of the predictions

Knowing the predicted basket, the following measures will be used for assessing the results.

Wasserstein distance: This distance will be used by computing the distance between the predicted basket and the real basket bought by the customer. As explained before, it uses vector representations of the words. This makes similar words such as 'white bread' and 'brown bread' to be mapped close together. In this way, it will not penalize too much the predicted baskets which contain similar products, rather than the exact ones in the actual basket.

F1-score: This metric is often used for accuracy testing, as it is the harmonic mean of precision and recall. Precision represents the number of accurate predictions out of the total predictions made. Recall refers to the number of correct predictions out of the total number of examples needed to be retrieved. In the context of market baskets prediction, the F1-score will be computed per basket. In this sense, precision will penalize the wrong predicted products, whereas recall will account for missing items in the predicted basket. In the end, the final score will be returned by taking the average of all the F1 scores of the predicted baskets.

Jaccard coefficient: The Jaccard metric is used for assessing the similarity between two sets. It is formally defined as the proportion between the size of the intersection of the two sets and the union of them. For two market basket X and Y, this translates to:

$$J = \frac{p}{p+q+r}, \tag{18}$$

where p represents the number of items which are included in both X and Y, q refers to number of items which are in X, but not in Y and r is the number of goods in Y and not in X. Same as $F1$ score, the final result will be obtained by taking the average of all the scores obtained per predicted market basket.

5 Results

As explained in the data section, a sample of the entire purchasing histories dataset has been considered. The methods have been ran on an virtual Ubuntu 18.04 with 8 vCPUs and 30 GB RAM.

Instacart Products

As Kraus and Feuerriegel(2019) report their results on the whole dataset, the correctness of this research's implementation has been compared to the original code[3]. It is also important to mention that the online appendix provided by creators of this method does not include the **similarity-aware prediction** or tunning the number of nearest neighbors, as explained in 4.4. Thus, the only hyperparameter in this part of the method is the dimension of the word representations as resulted from word2vec algorithm. The dimension is set to 50 for the rest of the results, as it gives an accurate representation of the words. By introducing the same number of purchase histories in the algorithm, after applying the filters, 1150 customers have been selected in this research implementation and 1260 in the original implementation. In both cases, 150 test baskets have been selected and the results obtained were similar. The detailed results can be found in Appendix B.

The following results presented refer to the full implementation of the algorithm presented in this paper which included a tunning process. For tunning τ and k (number of neighbors) hyperparameters, a validation set of customers has been further included. The k-nearest neighbors can take the any of the values $k = \{1, 5, 10, 15, 20\}$ and $\tau = \{5, 10, \ldots, 35\}$. The tunning results can be seen in Figure 3.

Table 1: Results obtained for Instacart product

	Wasserstein distance	F1 score	Jaccard coefficient
Global top products	9.94	0.151	0.087
Personal top products	6.982	0.387	0.262
Repurchase last basket	7.127	0.379	0.257
Method presented in this report	6.982	0.387	0.262

The classification has been made based on the F1 score. Thus, for this sample of the

[3]https://github.com/MathiasKraus/MarketBasket/tree/master/code

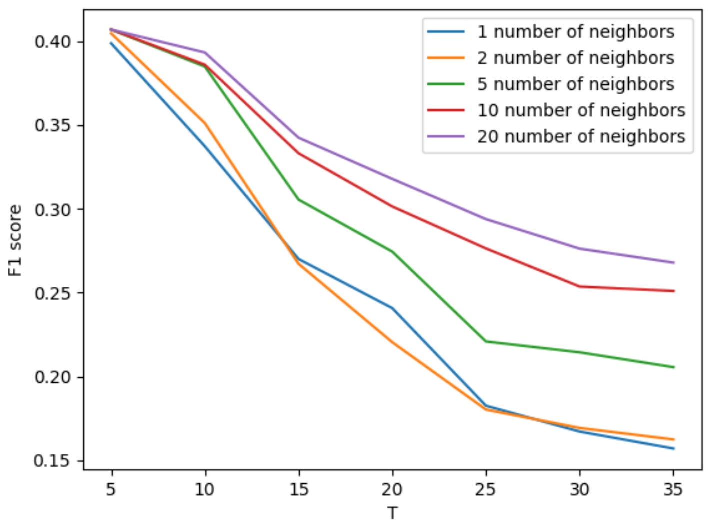

Figure 3: Tunning of k (nearest neighbors) and τ for Instacart product

dataset, the best combination of hyperparameters is setting the number of neighbors to any value except 1 and fixing τ to 5. This happens as a result of the lower threshold, most of the baskets being predicted with the fallback prediction. An extended presentation of the results can be checked in Table 1. It shows that for this reduced dataset, the results of our method equal the results of predicting the most common personal products, as all the baskets fall under the fallback prediction. This can be further investigated in Figure 4. This graph presents the ratio of baskets predicted through the considered heuristic for different pairs of parameters. For the validation set, the best scores obtained (which correspond to the pairs mentioned earlier) are when all the baskets are predicted through the personal top baseline, as the ratio equals 1.0. This baseline also represents the best method for predicting the next market basket in this version of the dataset, as all its prediction numbers are superior to the other methods.

Instacart Aisles

In the same manner as the previous dataset, the hyperparameters will be tunned. The detailed graphs can be investigated in the Appendix B.

In table 2 the results for this version of the dataset can be investigated. In the same way as before, the tunning process returns a pair of hyperparameters which favors the fallback prediction and, thus, all 100 baskets will be predicted with the personal top baseline.

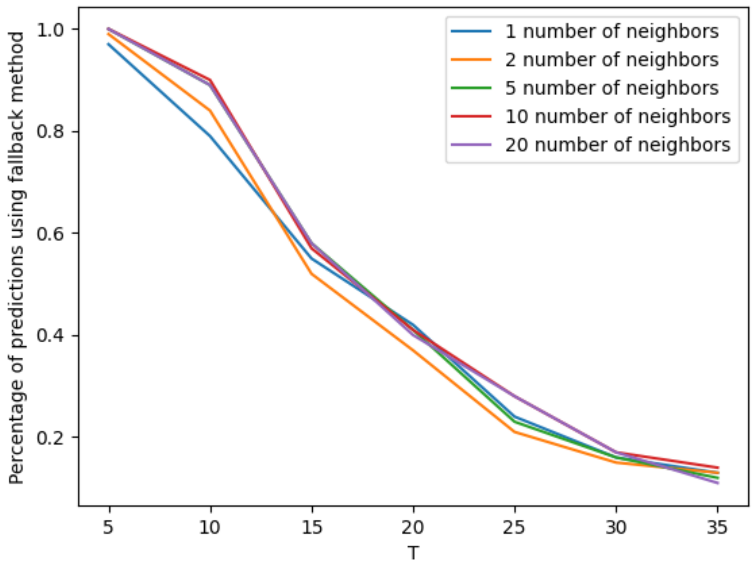

Figure 4: Ratio of next market baskets predicted by method for Instacart product

Table 2: Results obtained for Instacart aisle

	Wasserstein distance	F1 score	Jaccard coefficient
Global top products	4.819	0.398	0.33
Personal top products	4.302	0.496	0.452
Repurchase last basket	3.577	0.704	0.566
Method presented in this report	4.302	0.496	0.451

The validation run sets the number of neighbors to 20 and τ to 5 as for the product-level dataset. However, for this pair of hyperparameters, the fallback predictions are called only for 0.7 of all the market baskets predicted of the validation set. It is found, however, that the baseline methods outperform heavily the method proposed in this paper. In the case of Instacart aisles, the baseline of repurchasing the last basket performs the best, attaining a F1-score of 0.704. An idea to improve the algorithm would be assess which baseline performs best on the dataset given. Then, use this best baseline method for the fallback prediction.

E-Commerce As part of the extension of this study, it has been further considered to include a e-commerce dataset. The tunning of the parameters can be found at Appendix B.

Table 4 presents the results obtained when applying the method on this dataset, as well as the scores obtained by the considered baselines. It can be observed that the baselines methods of personal top products, as well as repurchase last basket perform quite similar to the method presented in this report. These two baselines outperform the algorithm by less than 0.01 when investigating the $F1$-score and Jaccard coefficient.

The percentage of baskets which fall under the fallback prediction is 83%, which is lower than before. Also, in a similar manner as with Instacart Aisle dataset, the resulting parameters are set to 20 number of neighbors and τ set to 5 as a result of the tunning process.

Table 3: Results obtained for E-Commerce

	Wasserstein distance	F1 score	Jaccard coefficient
Global top products	7.556	0.070	0.038
Personal top products	5.834	0.226	0.140
Repurchase last basket	5.834	0.227	0.143
Method presented in this report	5.910	0.221	0.135

The results for the samples of the two versions of the Instacart dataset are similar to the scores reported by Kraus and Feuerriegel(2019) when using the whole datasets. In the case of using the whole dataset, the algorithm manages to outperform by a small margin the baselines methods. This contrasts with experiments done in this research on the Instacart samples, which conclude that the baselines could be a better alternative than using the algorithm presented in this research. This fact is strengthened by results obtained for the additional E-commerce dataset which finds that the baseline methods perform slightly better than the algorithm proposed.

One of the main limitations of this algorithm is that it is computationally expensive to use. This contrasts with the facile baselines, which do not require many resources to compute the next market basket. Thus, in the case of limited computational power, using the 'personal top product' or 'repurchase last basket' baselines could be a more appropriate alternative.

6 Conclusion

The main research question referred to finding a method which uses both global and personal purchasing habits of the customers in order to predict the next market basket. In this sense, this research has replicated the methods of Kraus and Feuerriegel(2019) and an investigation has been made on the accuracy and efficiency of a different context than the one presented in their paper. The method has been tested on samples of two versions of the Instacart dataset, which was also used in the original paper. Then, as an extension, the method has been tested on an e-commerce dataset.

Following the algorithm proposed by Kraus and Feuerriegel(2019), the steps required for implementing this method have been presented. The first required step is to project the words onto a vector space such that similar products are close to each other. Then, the Wasserstein distance has been introduced for computing the similarity between two market baskets. Subsequently, dynamic time warping is used for computing the closeness of the purchase histories among different customers. This step was followed with a prediction scheme for the next market basket.

The results obtained by applying these methods on the considered samples could not lead to the conclusion that this method is superior to the baselines considered. The scores obtained are either equal or below the outcomes of returning the most popular products of a customer or returning the last basket a customer has bought.

Instead of analyzing the whole Instacart dataset, this research has only applied the methods on a small sample of the whole dataset due to a lack in computational resources, as well as time constraints. This shows an important limitation of the method referring to being computationally expensive. However, the method can still be applied in some contexts, such as direct marketing campaigns in which processing time is not necessarily an issue.

Thus, it is concluded in this report that for a small sample of market baskets applying one of the performing baseline methods represents a better alternative from both efficiency and accuracy perspectives. In this way, this research broadens the literature by presenting effective methods for limited computational resources.

In the future, the research could be extended in several ways. First, the computational resources could be expanded which would lead to the use of bigger datasets. Second, the prediction part containing the modified form of the k-nearest neighbors algorithm could be modified to a different one, such as using a recurrent neural network architecture. This network could be build by finding a preset number of most similar purchase histories, which would represent the features for predicting the next market basket of this individual.

References

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137–1155.

Crone, S. F., Lessmann, S., & Stahlbock, R. (2006). The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. *European Journal of Operational Research*, *173*(3), 781 - 800.

Guidotti, R., Rossetti, G., Pappalardo, L., Giannotti, F., & Pedreschi, D. (2017). Market basket prediction using user-centric temporal annotated recurring sequences. In *2017 ieee international conference on data mining (icdm)* (p. 895-900).

Guidotti, R., Rossetti, G., Pappalardo, L., Giannotti, F., & Pedreschi, D. (2018). Personalized market basket prediction with temporal annotated recurring sequences. *IEEE Transactions on Knowledge and Data Engineering*, *31*(11), 2151–2163.

Gupta, S., & Mamtora, R. (2014). A survey on association rule mining in market basket analysis. *International Journal of Information and Computation Technology*, *4*(4), 409–414.

Gutmann, M. U., & Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, *13*(Feb), 307–361.

Imran, M., & Alsuhaibani, S. A. (2019). A neuro-fuzzy inference model for diabetic retinopathy classification. In *Intelligent data analysis for biomedical applications* (pp. 147–172). Elsevier.

Kraus, M., & Feuerriegel, S. (2019, Jul). Personalized purchase prediction of market baskets with wasserstein-based sequence matching. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*.

Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015). From word embeddings to document distances. In *International conference on machine learning* (pp. 957–966).

Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems* (pp. 556–562).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Redbord, M. (2019). The hard truth about acquisition costs (and how your customers can save you.

Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on world wide web* (pp. 811–820).

Sakurai, Y., Faloutsos, C., & Yamamuro, M. (2007). Stream monitoring under the time warping distance. In *2007 ieee 23rd international conference on data engineering* (pp. 1046–1055).

Schroeder, B. (2018). If ecommerce retail revenue is only 10 of all global retail better understand these key strategies to take advantage of online sales.

Singhal, A., et al. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, *24*(4), 35–43.

Suparta, W., & Alhasa, K. M. (2016). Modeling of tropospheric delays using anfis.

Tran, N.-Q., Vo, B.-N., Phung, D., & Vo, B.-T. (2016). Clustering for point pattern data. In *2016 23rd international conference on pattern recognition (icpr)* (pp. 3174–3179).

Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S., & Cheng, X. (2015). Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th international acm sigir conference on research and development in information retrieval* (pp. 403–412).

Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th international acm sigir conference on research and development in information retrieval* (pp. 729–732).

Zimdars, A., Chickering, D. M., & Meek, C. (2013). Using temporal data for making recommendations. *arXiv preprint arXiv:1301.2320*.

A Artificial Neural Networks

An artificial neural network (ANN) is a supervised machine learning algorithm which mimics the processes of neurons in the human brain. It can be applied in classifying or regression type of problems. The most common use of this algorithm is in binary classification. An example of an application of binary classification is deciding whether a company should send a mail to a person or not, in order to maximise the chance that the person will respond to it (Crone, Lessmann, & Stahlbock, 2006).

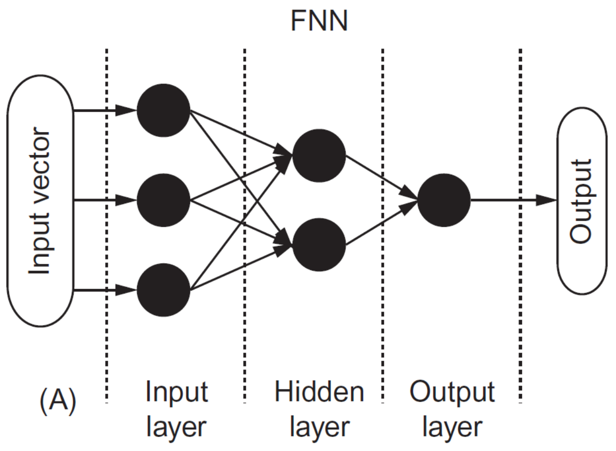

Figure 5: Feedforward ANN architecture, retrieved from (Suparta & Alhasa, 2016)

ANN considers each node as an artificial neuron, making it able to process nonlinear information (Imran & Alsuhaibani, 2019). The architecture of a neural network system contains an input layer, a hidden layer and an output layer, each containing a preset number of nodes. In the first one, the raw data is inserted, transferring it to the hidden layer. Each node in the input layer is connected to the hidden layer nodes, having asserted a certain weight. The hidden layer processes the information and then transfers it to the output layer. In this stage, the information is again processed in order to make the final inference.

Formally, each transfer from one node to another is associated with a cost. Thus, the aim of the algorithm is to minimise the sum of all the transfer costs. This can be written as follows:

$$y(k) = F(\sum_{n=1}^{N} w_i(k) \cdot x_i(k) + b) \tag{19}$$

,where $y(k)$ represents the inference of the algorithm, F is the activation function, $w_i(k)$ is the weight associated, $x_i(k)$ is the raw information and b is the possible bias. Possible forms of the activation functions are sigmoid or threshold-based.

B Results

Table 4: Results obtained for our implementation versus original implementation

	Wasserstein distance	F1 score	Jaccard coefficient
Our implementation	8.728	0.188	0.110
Original implementation	8.799	0.182	0.106

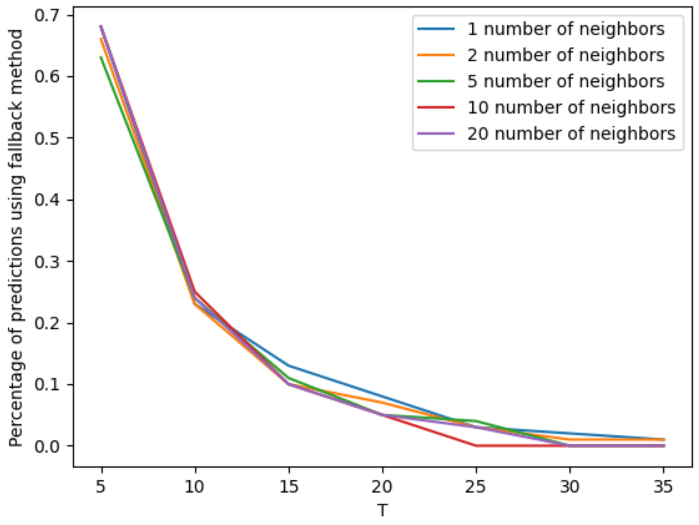

Figure 6: Ratio of next market baskets predicted by method for Instacart aisle

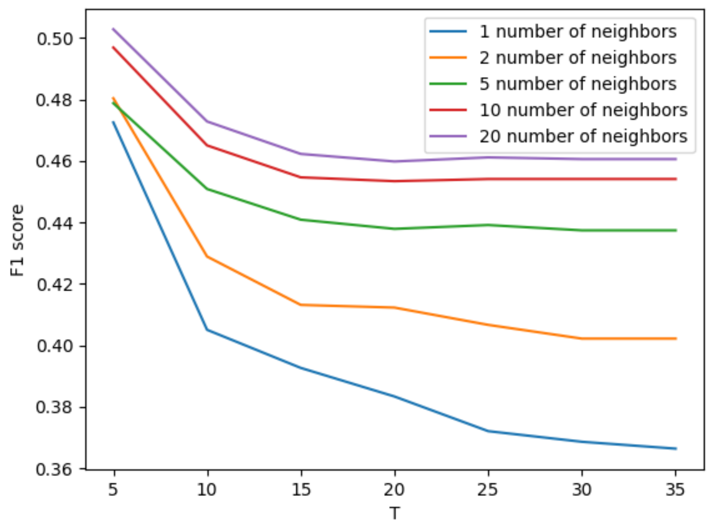

Figure 7: Tunning of k (nearest neighbors) and τ for Instacart aisle

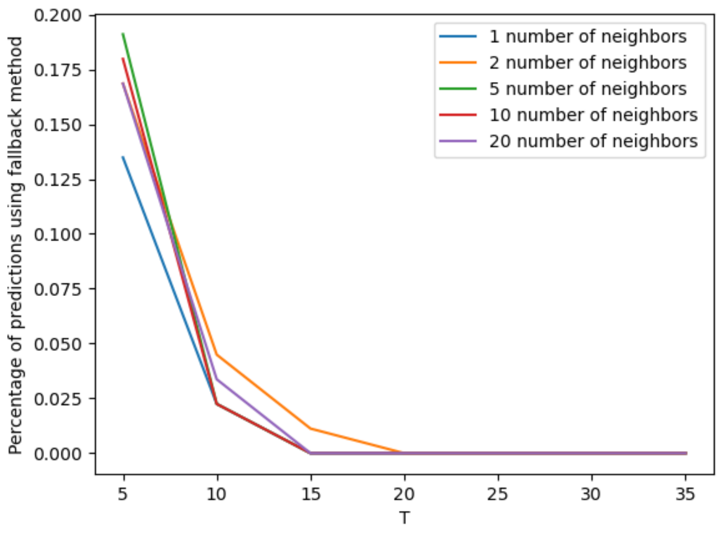

Figure 8: Ratio of next market baskets predicted by method presented for E-Commerce

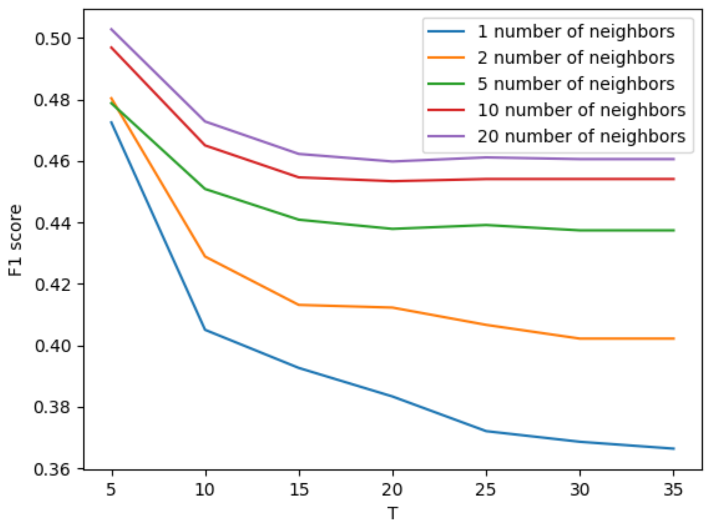

Figure 9: Tunning of k (nearest neighbors) and τ for E-Commerce dataset

C Code

The 'thesiszip.zip' contains the code and data required for the results presented in this paper. The code files have been compiled in Python 3. In the following, a short description of the python files is made:

anotherCreateData.py helps in creating the market baskets and filtering them according to the requirements presented in the data section.

anotherMain.py is the main part of the python code. Here, the class 'Distances' contains the lower bound and Wasserstein distance between two market baskets as well as the dynamic time warping algorithm mplementation. Furthermore, in the code, some parallelizing is applied for multiple cores processors. The code also contains the tunning done on the validations sets and the final results predicted through the test set.

itemEmbed.py is needed for creating item embeddings through word2vec algorithm.