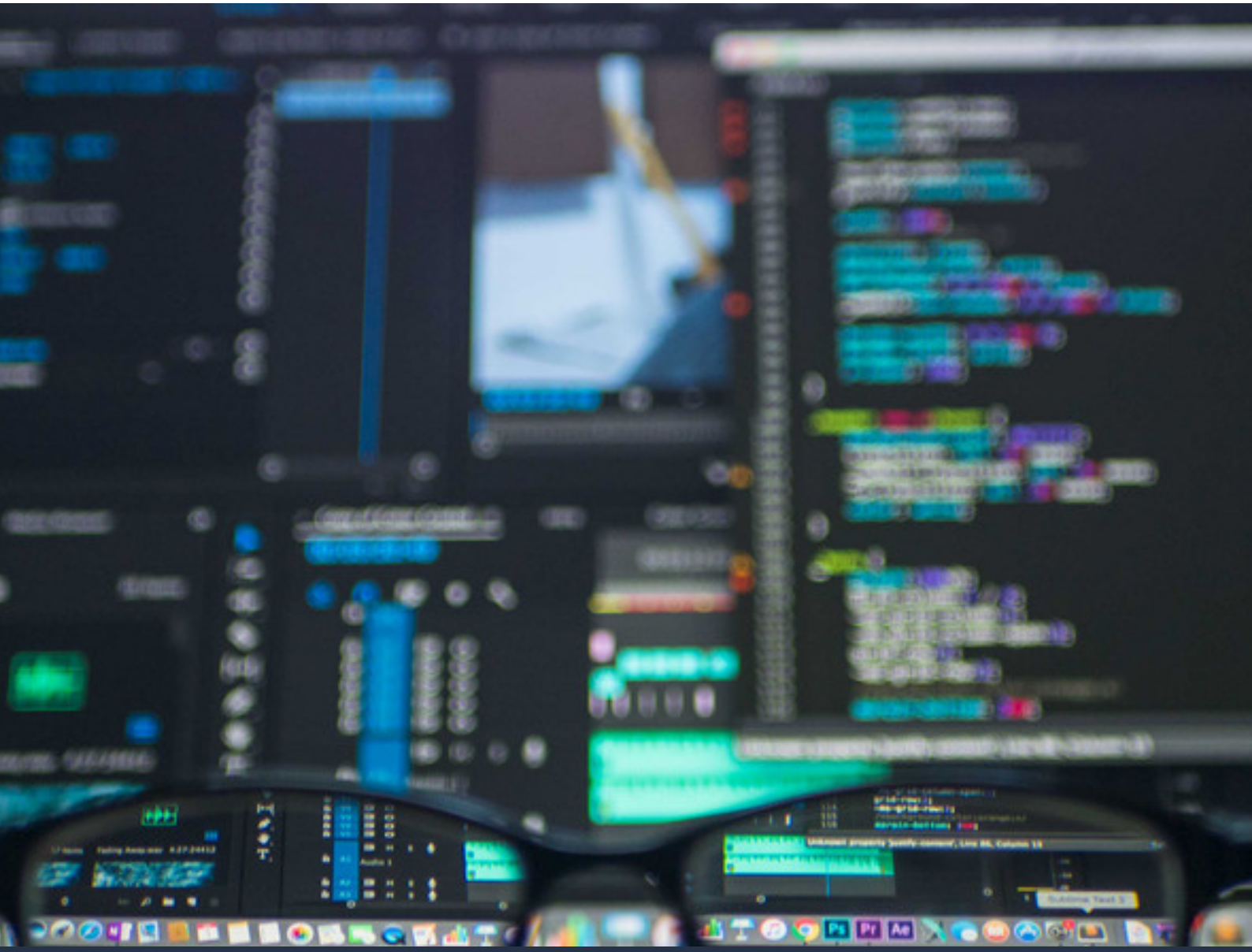
MAY, 2022

Algorithmic Approach

E-COMMERCE PLATFORM ANALYSIS - NTT PROJECT



Group members

Balkiss Tekaya, Giulia Di Martino, Martina Bozzi, Sirine El Feki, Tasnim Tekaya

NTT DATA

TABLE OF CONTENTS

Meet The Team

Balkiss Tekaya
Tunisian
Background in:
Business Administration
Business Analytics

Giulia Di Martino
Italian
Background in:
Finance

Martina Bozzi
Italian
Background in:
Political Science

Sirine El Feki
Tunisian
Background in:
Business Administration
Business Analytics

Tasnim Tekaya
Tunisian
Background in:
Business Administration
Business Analytics

GOALS OF THE PROJECT

•Customer Segmentation: Customer analysis that aims at identifying groups of users with similar purchasing behaviors in order to create a marketing strategy towards each target (Clustering)

•Recommendation System able to provide to each user a set of products related to their interests, in order to maximize the probability of purchase (Product Recommendation)

•Shipping time analysis to predict an accurate date of delivery and to reduce delays, in order to improve the quality of the shipping service (Delivery Time Prediction)

CUSTOMER SEGMENTATION

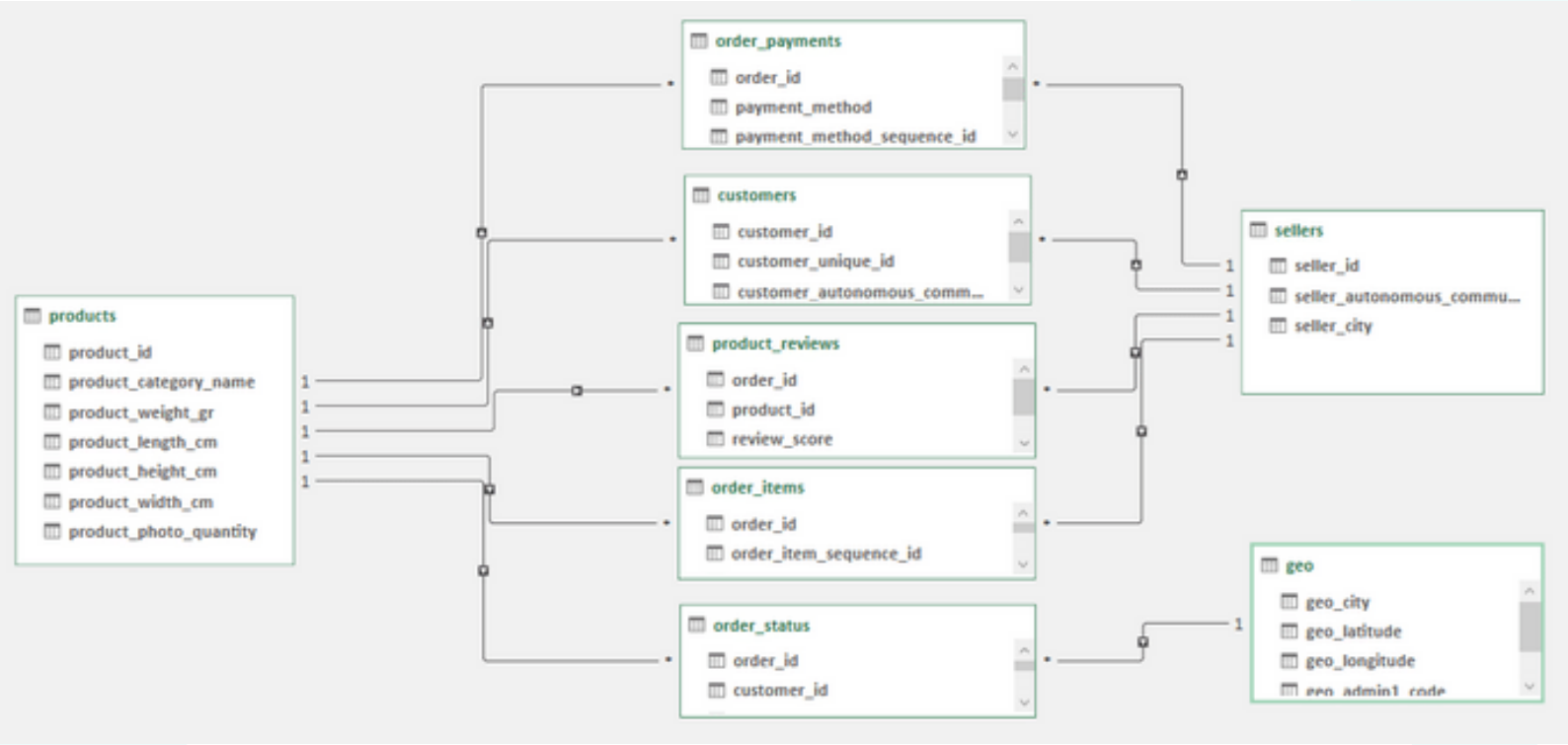
Data Preparation

Before starting any analysis, it is important to clean, preprocess and prepare a dataset that contains in each row a unique customer identifier, and each column some features describing that customer.
We start by cleaning and grouping each dataset (Table below) before we merge all the information for our final clustering analysis. Then, we merge the datasets (we make use of all the datasets except for suppliers.csv and geo.gsv)

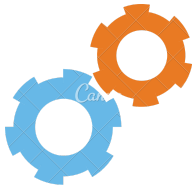
Dataset	Data Preparation	Group By
order_items.csv	* drop missing order_id (5) * drop columns 'max_shipping_seller_date' & 'seller_id' we don't need them in this analysis	Group by: order_id order_item_sequence_id: max() to get number of items per order product_id: join by "," if the products are different price: sum, shipping_cost: sum
order_status.csv	* only kept delivered orders * convert the variables to their appropriate type (float & date) * deleted ts_order_delivered_carrier & 'ts_order_approved * dropped NA	-
order_payments.csv	* remove NAs (only 12) * merged with a column with the unique customer ids	Group by: unique_customer_id 'payment_installments_quantity': 'max', 'payment_method': join by "," if the payments are different 'transaction_value': 'sum', 'payment_method_sequence_id': 'max'
product_reviews.csv	* merged with a column with the unique customer ids	Group by: unique_customer_id 'review_score': count() to get the number of reviews each customer made
products.csv	* we created a new category where we grouped further the categories	-
customers_csv	* only kept the variables 'product_category_name','category_new','product_id'	Group by: unique_customer_id

Merged Dataset

After we prepared our datasets, we merged them to get a final dataset that represents each unique customer id in a row and columns that describe him/her. We do the merging based on the relationships extracted from our entity relationship model as well as our specific needs.



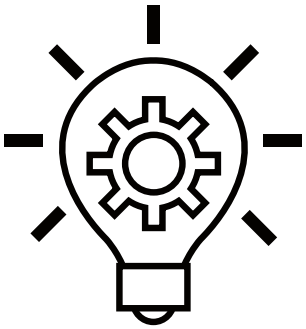
Feature Engineering



Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in Machine Learning.

Features	Description
Frequency	Number of orders per customer_unique_id
max_shipping	Maximum shipping cost is the maximum total shipping cost a unique customer spent per order
Tenure	Tenure is how long a customer used the app/website to purchase orders
Recency	How latest issue an X customer did, this is computed by taking the (latest date in the dataset)+1 and subtracting it by the latest date of each customer_unique_id
tot_ship/tot_spend	Total shipment spent per customer_unique_id over the Total_spending cost per customer_unique_id Intuition: the share of shipping cost compared to price of products purchased
new_diff	It is the maximum duration of estimated delivery from the each estimated wait per order_id (for a specific customer_unique_id)
Installment_payment	A binary variable; 0 for not paying is installment & 1 otherwise
del/rev_mean	The average time it takes for a customer to review bought product since the day of actual delivery
Payment_method	The payment method used per customer_unique_id; in this we can find a combination of method (e.g; Voucher, Credit Card)
tot_order_item_count	Total number of items bought by a unique customer
rev_count/tot_items_mean	Average number of reviews to the total items bought per unique customer

Modeling



To summarize:

In a first stance, a Principal Component Analysis (PCA) has been implemented in order to reduce the dimensionality of the dataset before carrying out the clustering. Dimensionality reduction allows to summarize the information content of large dataset in a smaller set of indices that can be more easily visualized and analyzed. In fact, the initial 14 variables have been reduced in 8 principal components.

Then the best number of clusters (segments) was determined by applying the Elbow method for K-means clustering. From the latter, it emerged that the optimal amount of clusters is 5 (k = 5).

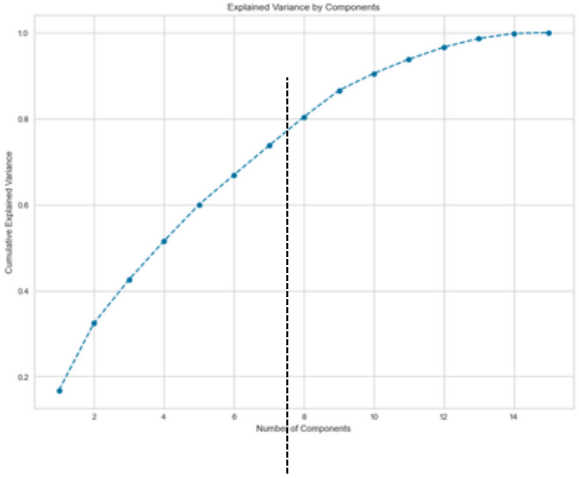
Three clustering Algorithms were tested for this analysis: Kmeans++, Gaussian Matrix, Birch.

Dimensionality Reduction

Principal Component Analysis

(PCA) is a commonly used technique to reduce the dimensionality of a dataset and eliminate the problem of multicollinearity and correlations between the variables. This is done through transforming the variables into a set of uncorrelated variables, called Principal Components, and are ordered in a way that the first components account for most of the variation of the data

In order to know the number of PCs to include, we use the screeplot method (Graph below). We look for the "elbow" in the graph and we adopt 8 PCs in out analysis.



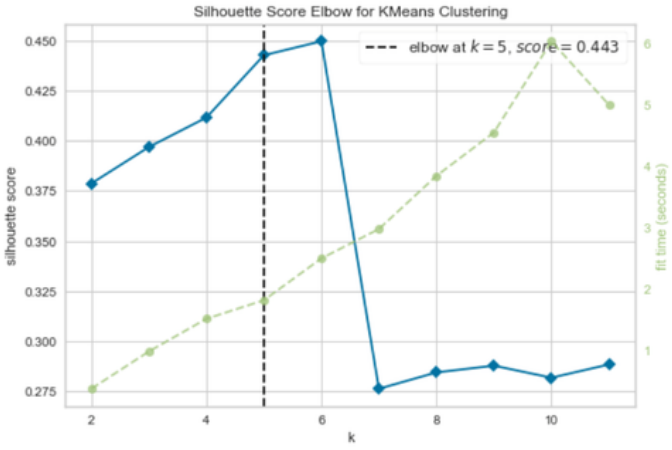
In multivariate statistics, a scree plot is a line plot of the eigenvalues of factors or principal components in an analysis.[1] The scree plot is used to determine the number of principal components to keep in a PCA.

Here we keep 8 PCs.

Choosing the number of clusters

Silhouette

A common heuristic of finding the best number of clusters is the silhouette method. The silhouette index is between [-1, 1]; the higher the value, the better the result. Hence, the most appropriate number of clusters according to this method should maximize the value of the index.



$$Silhouette = \sum_{i=1}^{n} \left[\frac{b(i) - a(i)}{\max\{a(i); b(i)\}} \right] / n$$

Where:
i: an observation in the dataset
a(i): average intra-cluster distance of i to other observations in the cluster
b(i): minimum average inter-cluster distance

Algorithms

Kmeans ++

Is an initialization technique of K-means which picks initial k-centroids using probabilities.

PSEUDOCODE

1. Choose one centroid uniformly at random from among the data points
2. For each point x compute D(x), the distance between x and the nearest centroid that has already been chosen
3. Choose one new data point at random as a new center, using a weighted probability distribution where a point x is chosen with probability proportional to D(x)2 .
4. Repeat Steps 2 and 3 until k centers have been chosen.
5. Now that the initial centers have been chosen, proceed using standard k-means.

Gaussian Mixture

Gaussian Mixture Models (GMMs) assume that there are a certain number of Gaussian distributions, and each of these distributions represent a cluster. Hence, a Gaussian Mixture Model tends to group the data points belonging to a single distribution together.
Objective function of GMM is to maximize the likelihood value for the data X, p(X) or the log-likelihood value L(since log is a monotonically increasing function). By assuming a mixture of K gaussians to have generated the data, we can write p(X) as marginalized probability, summed over all K clusters for all data points.

BIRCH

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is a clustering algorithm that can cluster large datasets by first generating a small and compact summary of the the large dataset that retains as much information as possible. This smaller summary is then clustered instead of clustering the larger dataset.

Choosing the best model

We use the silhouette score to select the best algorithm.
We calculate this score for the three adopted algorithms and we compare the scores to select the one with the highest value.
We find that Kmeans ++ is the best algorithm according to this criteria with silhouette = 0.44.

RECOMMENDATION SYSTEM

Market Basket Analysis

Market basket analysis is a data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together.
Association Rules are widely used to analyze retail basket or transaction data, and are intended to identify strong rules discovered in transaction data using measures of interestingness, based on the concept of strong rules.
We conduct our analysis on categories instead of products because they are interpretable and not hashed.

Data preparation

- Grouped by order ID
- Combined product ids & category names joined by ","
- Sum of price & shopping cost
- Perform One hot encoding
- Create a matrix where cell = True if two categories have been purchased together

Results

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(fashion & shoes)	(Other)	0.025543	0.079183	0.011494	0.450000	5.683065	0.009472	1.674213
1	(Other)	(home accessories)	0.079183	0.148148	0.017880	0.225806	1.524194	0.006149	1.100309
2	(fragrance)	(beauty & personal care)	0.035760	0.091954	0.015326	0.428571	4.660714	0.012037	1.589080
3	(kids)	(comics)	0.118774	0.085568	0.025543	0.215054	2.513240	0.015379	1.164961
4	(comics)	(kids)	0.085568	0.118774	0.025543	0.298507	2.513240	0.015379	1.256216
5	(home accessories)	(furniture)	0.148148	0.263091	0.030651	0.206897	0.786408	-0.008325	0.929147
6	(furniture)	(kitchen & dining)	0.263091	0.260536	0.089400	0.339806	1.304255	0.020855	1.120070
7	(kitchen & dining)	(furniture)	0.260536	0.263091	0.089400	0.343137	1.304255	0.020855	1.121862
8	(lawn garden)	(furniture)	0.093231	0.263091	0.021711	0.232877	0.885158	-0.002817	0.960614
9	(tools home improvement)	(furniture)	0.063857	0.263091	0.030651	0.480000	1.824466	0.013851	1.417133
10	(kids)	(toys games)	0.118774	0.065134	0.024266	0.204301	3.136622	0.016529	1.174899
11	(toys games)	(kids)	0.065134	0.118774	0.024266	0.372549	3.136622	0.016529	1.404454
12	(wellness & relaxation)	(kitchen & dining)	0.063857	0.260536	0.054917	0.860000	3.300882	0.038280	5.281883
13	(kitchen & dining)	(wellness & relaxation)	0.260536	0.063857	0.054917	0.210784	3.300882	0.038280	1.186169

Example (rule 12) : if a customer purchases from category wellness & relaxation, they have 86% chance to buy from kitchen & dining.
So we can perhaps recommend products from this category !

RECOMMENDATION SYSTEM

Singular Value Decomposition

The **Singular Value Decomposition** (SVD) is widely used as a collaborative filtering technique. It uses a matrix structure where each row represents a user, and each column represents an item. The elements of the matrix are the ratings that are given to items by users. The aim for the code implementation is to provide users with item' recommendation from the latent features of item-user matrices.

Data preparation

- Grouped by order ID, Customer Unique ID & Product ID
- Checked for duplicates in customer_unique_id & product ID ie customers who bought the same product two different times & same rating
- Filtered the dataset keeping on those who reviewed at least 2 products and products that were reviewed by at least 5 users.
- Kept only the the customer_unique_id, product_id and review_score in the matrix.

Results

Since the results are represented as hash codes of the product_id, we couldn't infer much sense out of the result. However, reflecting on the categories of these products it output interesting results based on customer similarity in purchasing products we found for example an item in "Fashion & Shoes" recommended with an item in "Beauty & Personal Care".

SHIPPING TIME ANALYSIS

Exploratory data analysis

The datasets used for the following analysis are the following: customers, geo, sellers, items and order status. Consequently, the datasets have been merged in one unique dataset named "time_shipment" in which new variables were computed. The latter are:

- actual_wait" : delivery date to customer - day of the purchase
- "estimated _wait": estimated delivery date- day of the purchase
- "diff": the difference between estimated and actual wait to get an insight of the over/under estimations of the latter
- "distance": computed using latitude and longitude of customers and sellers with the Harvesine formula
- "op_carrier": difference between when the order was delivered to the carrier, and when it was purchased.

From the exploratory data analysis the following information emerged:

- On average the time for delivery is of 12 days,
- The company's estimated wait is of 23 days,
- 11 days over-estimation.

Data preparation

In this phase of the analysis, a new dataset has been created. The latter includes just the variables needed for the shipping time analysis.
The outcome variable chosen is: actual wait.
The independent variables used for modeling are the following:

- number of items,
- number of products,
- number of sellers,
- op carrier,
- maximum distance.

Modeling

To summarize:

In order to predict an accurate date of delivery and reduce the delays, three models were fitted for the shipment time analysis.

The models are:

- K-nearest neighbors ,
- eXtreeme Gradient Boosting,
- Random Forest.

The choice of the model relies on some typical features such as accuracy, simplicity, performance and interpretability. In this case an accurate model is needed, a one-shot model which provides accurate prediction.

Algorithms

K-nearest neighbors

K-NN is a supervised machine learning algortithm which relies on labeled data to learn a function that produces an appropriate output when given unlabeled data.

How it works:

- **Load the data**
- **Initialize K to your chosen number of neighbors**

Then for each example in the data:

- **Calculate the distance between the query example and the current example fromt he data**
- **Add the distance and the index of the example to an ordered collection**
- **Sort the ordered collection of distances and indices from smallest to largest**
- **Pick the first K entries from the sorted collection**
- **Get the labels of the selected K entries**
- **If a regression, return the mean of the K labels**
- **If classification, return the mode of the K labels**

To select the K that's right for your data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before.

The main advantage of this algorithm is that it has a good interpretability. A major disadvantage of this algorithm is that it gets significantly slower as the number of examples and/or predictors/independent variables increase.

eXtreeme Gradient Boosting

Extreme Gradient Boosting is mostly used because is a fast and accurate algorithm which handles missing values.

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of decision trees.

The main feauture of XGBoost is that it uses a more regularized model formalization to control over-fitting, which gives it better performance. Indeed, regularization helps to smooth the final learnt weights to avoid over-fitting.

Additional characteristics are:

- Gradient tree boosting: the model is trained in an additive manner,
- Shrinkage and column subsampling: 2 additional techniques to prevent over-fitting.

Random Forest

Random Forest algorithm is a great when high accuracy is needed, with less regard for interpretation.
Some of its main advantages are:

- good at learning complex and non-linear relationships,
- very easy to interpret and understand,
- it is accurate

Some of its disadvantages are:

- interpretability is not good,
- performance is not that good: using larger random forest ensembles to achieve higher performances slows down their speed and then they also need more memory
- a lot of tuning is needed, same as the cart tree

The steps required to carry out a Random Forest Regression are the following:

1. Pick at random k data points from the training set,
2. Build the decision Tree associated with this K data point,
3. Choose the number Ntree of trees you want to build and repeat STEPS 1 & 2.,
4. For a new data point, make each one of our Ntree trees predict the value of Y for the data point in question and assign the new data point the average across all of the predicted Y values.

Results

Since the necessity was to find a model able to predict an accurate date of delivery and reduce delays in order to improve the shipping service, among the three fitted models, the one found to be the most accurate has been the Random Forest model. The latter misses of 5 days compared to the 11 days initially estimated by the company. Indeed, in this way there is an increment in the delivery time prediction of more than 50%.