Haute Ecole ICHEC - ECAM - ISFSC

The stock replenishment through the machine learning

Graduation work presented by

Hadrian and Hachez

In preparation for the graduation of

Master's degree in Industrial Engineering Sciences with a computer science orientation

School year 2019-2020

Certificat TFE

Par la présente, nous certifions que Hadrien Hachez, né à Etterbeek le 9 mars 1993 et domicilié(e) à l'Avenue des héliotropes 2a 1030 Bruxelles, a bien effectué son travail de fin d'études en partenariat avec Avanade durant l'année académique 2019-2020.

Fait à Bruxelles, le 25/05/20

Signature et cachet responsable entreprise

Evelyne Meiresonne

Avanade Belgium bvba
Guldensporenpark 76 Blok H
B-9820 Merelbeke
Tel: +32 9 272 91 00
Fax: +32 9 272 92 00
BTW nr: BE 0480 059 829

CAHIER DES CHARGES RELATIF
au TRAVAIL DE FIN D'ETUDES de

Hadrien Hachez inscrit(e) en 5ème informatique

Année académique : 5EI

Titre provisoire :
Optimization of stock replenishment through machine learning and Microsoft Azure environment.

Objectifs à atteindre :

- Replenishment of stock by automated orders.
- Estimation of future sales.
- Algorithm developed on Databricks Azure in connection with Dynamics 365.

Principales étapes :

- Understand how the company operates.
- Understand the different predictive machine learning algorithm.
- Deploy and train the appropriate algorithm on Databricks on a real data set of an existing company
- Improve the sensitivity of the algorithm with external data known to be a factor in sales.

Fait en trois exemplaires à Bruxelles, le 18/11/2019

L'Etudiant	Le Tuteur	Le Promoteur
Nom-prénom : HACHEZ Hadrien	Nom-prénom : COMBÉFIS SÉBASTIEN	Nom-prénom : Tack, Patrick
	Département/Unité ....GEI...............	Société ....AVANADE
Signature	Signature	Signature

Table of contents

Abstract 6
Introduction 7
Objectives 8
Covid-19 and its impact on graduation work 8
Structure 9
Microsoft Azure 10
Introduction 10
Sample Program Structure 11
Azure Data Factory 11
Azure Data Lake 12
Azure Databricks 12
Azure machine Learning services 12
Handling of Azure tools 12
Databricks vs Azure Machine Learning Studio 13
Azure Machine learning Studio 13
Floris Van Bommel 15
Structure and organization 15
Product and client 15
Generalization and first analyses 16
Stock management 17
The data at your disposal 17
Data Profile 18
Research tracks 19
Recurrent neural networks 20
Introduction 20
Types 20
Operation 21

Problem and origin of LSTM style algorithms ... 22
LSTM ... 23
Case study in the case of stock management ... 24
Time series ... 25
Introduction ... 25
Model ARIMA ... 25
Practical case of the thesis ... 26
Model improvement and conclusion ... 29
Convoy management ... 31
Introduction ... 31
Algorithms Knapsack ... 31
Simulation vs reality ... 31
Sub-stock ... 31
Overstock ... 33
Conclusion ... 33
Acknowledgements ... 35
Conclusion ... 36
In the depths of the difficulties of learning machines: paradoxes ... 37
The Simpson’s Paradox ... 37
The Braess’s Paradox ... 37
Moravec's paradox ... 38
The accuracy paradox ... 38
The paradox of learning ability ... 39
Sources ... 41

Abstract

In 1959, the American computer scientist Arthur Samuel first used the term "machine learning" for his self-taught draughts program. This field of study of artificial intelligence is based on mathematics and statistics to give a machine the ability to learn on the basis of the data provided to it.

Sixty years later, this ubiquitous discipline appears to be fundamental in an increasingly dynamic and complex world. The commercial sector is a perfect example of this: faced with an increasingly demanding clientele, logistics must adapt. In the field of stock management, transport optimization and the preparation of future orders, the predictive nature comes into play and requires rigorous analysis.

This thesis proposes an analysis of the state of the art of machine learning about the stock and sales management of a trading company, and more precisely on its ability to predict future sales based on the data available to it. It compares the different algorithms currently in use, establishing their respective use cases and their limits. It also highlights the uncertain nature of prediction, the limitations of available data, and the complexity of the world around us.

Keyword: Stock Replenishment, machine learning, LSTM, CNN

Introduction

This thesis work was carried out under the aegis of Avanade, a consultancy firm in partnership with Microsoft and Accenture. Avanade wants to be a complete company that provides a service tailored to each of its customers. Using tools provided by Microsoft, solutions are implemented to meet the customer's needs.

Avanade's challenges are twofold: on the one hand, there are common business problems that Avanade is used to dealing with but whose optimal solution can evolve with the new tools that Microsoft offers. On the other hand, there are issues that the consulting company has never encountered but would still like to learn how to deal with. A preliminary research work is then undertaken, and solutions can thus be proposed to the client. If the client is interested and agrees to go further, a budget is made available.

The potential client at the origin of this thesis is the company Floris Van Bommel, a company that creates its own models of shoes and takes care of distribution and sales. Initially, Avanade is to help this company for better data accessibility, whose interface is on Dynamics 365. But as part of this thesis, research work to try to automate stock management and redistribution was initiated.

Indeed, in the world of commerce, a recurring problem is to be able to have at the disposal of the customer the product he wants to buy, in other words to have the right product at the right place at the right time. Two problems arise from this objective: overstock and shortage. Overstocking consists of overestimating sales and thus increasing the cost of storage unnecessarily. Shortage, on the other hand, is the underestimation of sales at certain points. Very often, there are variations in stocks in the various points of sale and it is for this reason that transport is generally organized between the points of sale and warehouses in order to avoid the 2 problems mentioned above locally.

This thesis is therefore intended to be the creation of a program capable of establishing and managing the necessary stocks in the different stores of a sales company. The algorithm focuses on the one hand on the prediction of the necessary stocks in each of the stores, and on the other hand on the optimization of the convoys.

Since the client company refused to make the data available for this project, the development of the algorithm was not able to go as far as desired. Historical data will have been used to analyze the quality of the learning techniques of the program, but the program will not have been able to influence any future sales as initially desired. The consequences of an algorithm unable to influence the future will also be developed in this work.

Objectives

Initially, the thesis wants to propose a theoretical algorithm capable of managing the stocks of a company based on an estimate of sales and of on an optimization the transports for the balancing of its last ones between the various points of sales and points of replenishment.

It starts with the focus on the structure of the Van Bommel company and its operation. It will then look at the type of product that the company offers and at the elements allowing the classification of these products. Based on the data provided. She will then try to predict the stock needed in each of the stores based on the data collected in the past and present by researching the learning techniques used in Machine Learning. The thesis will finally focus on transport and its optimization.

Covid-19 and its impact on graduation work

The end-of-study work is spread over two quarters. As I didn't have classes until Christmas, I was able to invest myself fully in this challenge, working full-time between October and mid-January.

The first month was a recovery of Microsoft Azure, its tools and access to its latest technologies.

The two following months were very busy: negotiations for data access (I was able to get access to a virtual environment with fictitious data, but these were very poor in data), research work on the large data scientist approach on this subject, understanding different notebooks and learning algorithms. At the beginning of 2020, the thesis started to take shape, while waiting for access to Van Bommel data for model training. Once I knew that I would never have access to the data, I finished my dissertation by adding examples of use cases of the algorithms I would have liked to use. I also took a step back and took elements of inventory management from other companies (such as Decathlon where I worked as a student salesman in September 2019) to illustrate the difficulties of a model.

Given the maturity I had gained and the methodology I had chosen to work on this thesis, I decided to still finish the writing and present the work in June.

The covid-19 thus had no direct impact on the work itself, only on the difficulty of obtaining signatures for the certification of the thesis.

The objectives to be achieved were the following:

1. Stock replenishment based on automated orders.
2. Estimates of future sales
3. Algorithm developed on Databricks in connectivity with Dynamics 365

Since Van Bommel could not provide us with the actual data, these 3 points had to be adapted as follows:

1. Understanding and handling of the Azure tools, with a view to a future implementation of a machine learning algorithm.
2. Learning and familiarization with algorithms useful for automated stock replenishment
3. Highlighting the undeniable factors of sales that are difficult to model in the learning machine.

Structure

This document will be divided into 5 distinct parts.

A first part will identify the tools put forward by Microsoft to allow its users the best possible ergonomics in their work.

A second part, a detailed understanding of the functioning of the company Floris Van Bommel as well as the objectives and problems encountered during the end of the study. An analysis of the organization will be carried out, the profiling of the types of customers as well as the product sold.

In the next part, the stock management through the learning machine will be criticized. In particular, the difficulty of a purely observational algorithm, the lack of data for modelling and the unpredictability of the economic world will be highlighted.

Before the conclusion, a study on the algorithms capable of solving the knapsack problem will be highlighted, at least if it is necessary.

The thesis will finally conclude, offering a last feeling about the machine learning and the difficulties it adds to the understanding of the business.

Microsoft Azure

This part will be a brief overview of Microsoft Azure.

In the context of this thesis, the deployment part will finally have been neglected compared to the training of the model. On the one hand because this deployment will have already been done as part of an internship for another model. On the other hand, because external and unpredictable difficulties have slowed down the progress of this thesis.

After a brief introduction of the potential of Azure, the tools envisaged for the deployment of the model will be put forward to allow the user to better understand the solution proposed by Azure for the training and deployment of Machine Learning solution.

Introduction

Microsoft Azure is a complete cloud application platform, allowing complete or partial hosting of a company's data and functionalities. This tool wants to be able to transform the ideas of its user into solutions and this through Azure products and services.

Microsoft offers complete hosting (applications and data) and services (workflow, data storage and synchronization, message bus, contacts...) through its web portal Portal Azure[1]. This set of APIs allows an easy use and accessibility to this platform and associated services.

Microsoft also offers "Azure (Remote) PowerShell" for more specific and granular actions that would require a special adaptation of the code.

The Live Operating Environment, a runtime environment, allows close integration with the various existing operating systems, namely Windows, Mac OS and Windows Phone.

The Windows Azure platform corresponds to Microsoft's PaaS[2] and IaaS public cloud computing offerings.

Among the elements of which it is composed, we can find in particular:

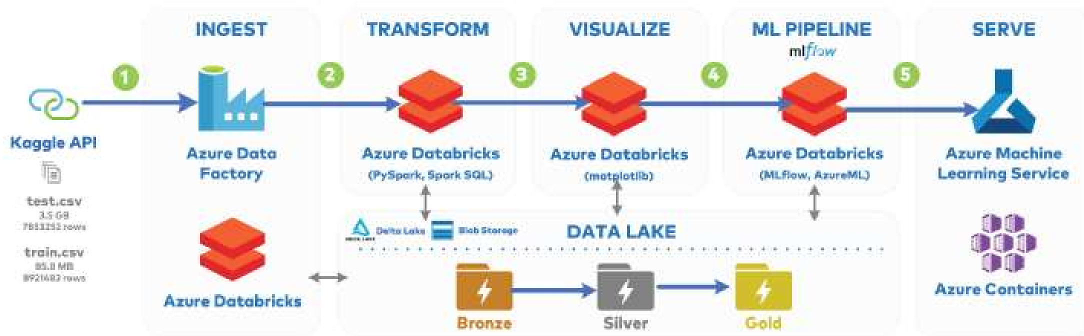
- WebApps (PaaS)
- Application roles (cloud services, such as PaaS)
- Windows Server or Linux virtual machines
- The virtual network (IaaS)

[1] https://portal.azure.com
[2] Cloud Computing model, as well as SaaS, DaaS and IaaS. A cloud service provider offers hardware and software tools as a service via the Internet, allowing the user to develop applications.

- Windows Azure storage (blobs, tables, queues, file sharing, drives)
- Premium storage attached to virtual machines

In addition, Windows Azure SQL database (formerly called SQL Azure) is a relational database server. The databases are thus used as a service.

Sample Program Structure



The data history can be brought into Azure in two different ways. Either through an API which will require a tool such as the Azure Data Factory for ingestion, or through a table stored in a csv file which can be opened directly in Azure Databricks.

The data preparation and transformation phase are done in Databricks, as well as the training phase of the model.

The deployment phase, which therefore allows the use of the model on a large scale, requires a certain control and supervision of the decisions made by the model. This is done at the level of the Azure Machine Learning Service.

During the preparation and training phase, backups of the different models are saved in a Data lake to allow deployment and possible reversal.

Azure Data Factory

Azure Data Factory allows you to integrate data silos. This service, designed for all data integration needs and skill levels, lets the user easily create ETL[3] and ELT processes with or without code in an intuitive visual environment.

[3] Extract, Transform, Load

Azure Data Lake

Azure Data Lake is a scalable data storage and analysis service. The service is hosted in Azure, Microsoft's public cloud. Users can store structured, semi-structured or unstructured data produced by applications including social networks, relational data, sensors, video, web applications, mobile and desktop devices. A single Azure Data Lake Store account can store trillions of files that can exceed petabyte in size.

Azure Databricks

Azure Databricks allows first Big Data analysis and the development of artificial intelligence in an optimized Apache Spark environment. This recent tool on Azure allows to acquire insights from data and to create intelligent programs. By configuring the Apache Spark environment, Databricks allows the user to adapt the power to the requested workload through automatic scaling, thereby optimizing the cost of the program. Designed for everyone, Azure Databricks supports languages such as Python, Scala, R, Java, and SQL, while integrating data science infrastructures and libraries such as TensorFlow, PyTorch, and Scikit-learn. Through its clusters and notebooks, Azure Databricks makes it easy to manage, monitor and update Machine Learning models deployed from the cloud to the edge.

Azure machine Learning services

Although Azure machine services allow the creation, testing and management of learning machine models, this service is in our case mainly used for its ability to migrate or monitor the models in an environment. This service allows its user to take care of the deployed models and monitor the different executions to get the best result, and this with asynchronous predictions (in real time) on a massive amount of data.

Handling of Azure tools

The Azure tools each have a documentation allowing an easy handling. This documentation can be found on this page[4].

This was not the first time I used this documentation for the simple reason that I had already completed a 3-month internship at Avanade.

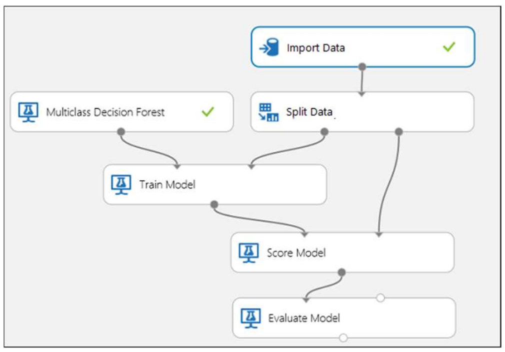
[4] https://docs.microsoft.com/fr-fr/azure/?product=ai-machine-learning

It gave me an overall feeling. Portal Azure is like a jigsaw puzzle: at first, you can feel lost because there are too many elements to consider, too many dependencies between each program. But the further you go, the faster it ends. And just like a puzzle that you do repeatedly, the same structure in Azure becomes easier and easier to build.

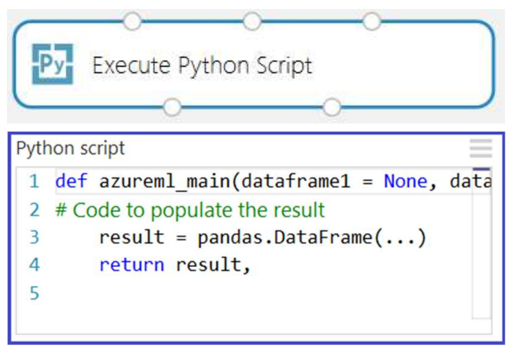
Databricks vs Azure Machine Learning Studio

In this part

Azure Machine learning Studio



Microsoft Azure Machine Learning Studio (classic) is a drag-and-drop collaborative tool that allows you to generate, test and deploy predictive analytics solutions from your data. Azure Machine Learning Studio (classic) publishes models as web services that can be easily consumed by custom applications or business intelligence tools such as Excel.



My feeling about this program at the end of my internship was that it was a great tool for people who want to make simple algorithms without too much programming. Indeed, the drag-and-drop aspect is simply easy to handle and control. However, when one wanted to make more specific and complex algorithms that require several adjustments, lines of code became mandatory. ML studio proposed boxes in which lines of code could be written, but this solution quickly became futile as the graphics became difficult to follow and control.

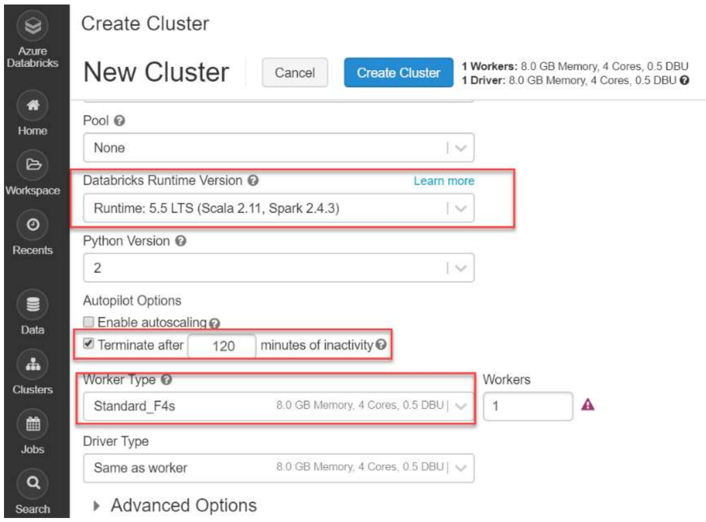
Since my internship, Azure has added other tools, including Databricks. The goal of this thesis was to discover this tool, that's why I don't take too much time to dive into the documentation. Databricks is the tool, in my opinion, that all computer scientists using Azure were waiting for. It is easy to use.

Its documentation[5] is clear, but I only needed it briefly because once I was in the notebook, I was in my element. A simple notebook framed by a cluster that can be adjusted. A computing power that

[5] https://docs.databricks.com/

allows an analysis of the first trainings in only a few moments, where on a classic pc, it would take much more time.

The only negative point was that it looked so much like a classic environment, that for a thesis like this one, very quickly I felt that I could run the same work for free without the execution time being a problem.



I think Databricks will be the tool I will use, because of its similarity with the tools I am used to use, but also because of its compatibility with other Microsoft Azure software.

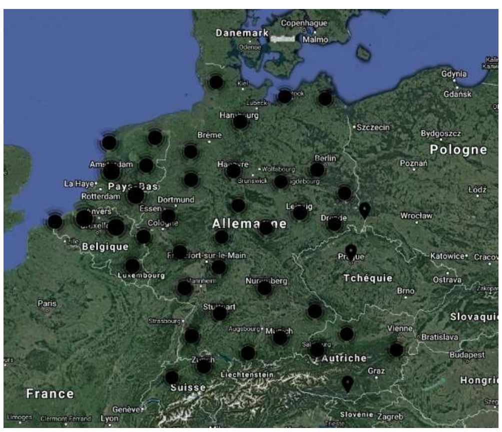
Floris Van Bommel

This first part focuses on the company Floris Van Bommel, and more specifically on its structure and organization, the products offered for sale and the target customers.

Structure and organization

Van Bommel is a Flemish company that produces and sells its own shoes.

Its origins date back to 2008, when Van Bommel opened its first store in Antwerp, Belgium. At the beginning of 2016, the company starts online trading, enabling its customers to have their shoes delivered to their homes. By 2019, the company has two stores in Belgium, five in Germany and four in the Netherlands. But to increase the number of points of sale, the company is also using stores outside the company by allowing them to sell and display Van Bommel brand shoes in their shop windows. This allows the company to cover many territories such as Belgium, the Netherlands, Germany, Austria, the Czech Republic and Switzerland.



Initially, each store has an Excel spreadsheet with all the shoe models at its disposal as well as a sales history. The company's official stores also provide the customer with a loyalty card. The supply depends on the point of sale but is generally carried out every two days on the basis of the orders recorded by each store and according to availability. It is the company itself that chooses the models that will be promoted in each store but it is the store that decides which stock it wishes to have.

Every year, Floris Van Bommel designs new shoe models himself and decides which models will no longer be restocked.

Product and client

The products offered by the company are luxurious and elegant, with prices ranging from €125 to €290[6] depending on the desired model. As the models are very varied, the store mainly aims to provide shoes for both men and women. On the site itself, a series of criteria is already proposed to the

[6] Prices as of May 2020 on https://nl.florisvanbommel.com

customer to classify the shoes. Men's models, women's models, the type of shoe (slippers, loafers, moccasins, ...) the size, the material, the color, the width, the character (sporty, casual, premium, ...), the type of sole, as well as the texture (matt or shiny).

Generalization and first analyses

This section focuses on the generalization of the Van Bommel case to other companies. A company generally consists of several sales outlets, either exclusive or partial (depending on whether the store is owned by the company), and one or more warehouses where the shoes are shipped once the design is completed and checked.

The backpack problem seems to be a first challenge for an algorithm that would seek to automate the supplying of stores from the warehouse, balancing a store that is unable to exhaust its stock to a store that lacks stock, and finally a model that is no longer sold in the store and must be returned to the warehouse. Both temporal and geographical organization is a real challenge. How often should replenishments be carried out, in what order should stores be stocked, etc.?

A second challenge appears, that of stock prediction. During the same season, each store has to ask itself what its future sales will be and therefore the stock to forecast. In the case of shoes, these are determined by a model and a size. These two criteria are important because they allow a better classification and adjustment of stocks.

A final challenge, probably the most difficult, would be to predict sales during disruptions. If a model were to run out, what would be most likely to replace it? What impact will the new season have on the old models in terms of sales (which models will be most affected)?

In addition, for each of these problems, the first difficulties appear. Do all orders have the same weight, if not how to classify them? Can an inventory error be negligible for the training of a model? Can the events of the past really be comparable with those of today, is precisely in the world of commerce, dictated by fashion and the desires of the customer? What are these disruptive elements that can influence sales, are they measurable quantities, or rather a subjective feeling? In the rest of this thesis, we will try to answer these questions and analyze the methods commonly used to approach reality.

Stock management

This chapter will be devoted to an in-depth study of the algorithm chosen to deal with stock management.

Because the data of the company Van Bommel could not be accessible for this thesis, an external database[7] available on Kaggle was used. Please note that there will obviously be differences with Van Bommel's data (not the same products, not the same time period) the development below should be illustrative.

However, the task remains the same: forecasting sales in the stores of a chain store to avoid overstocking, which would reduce waste and losses, and to minimize understocking, which leads to opportunity cost and lower customer satisfaction. In the interest of both business and the environment, better forecasting is highly desirable. This increased efficiency could also translate into higher benefits for stakeholders and/or a better price for customers, depending on the choices made by the chain.

The data at your disposal

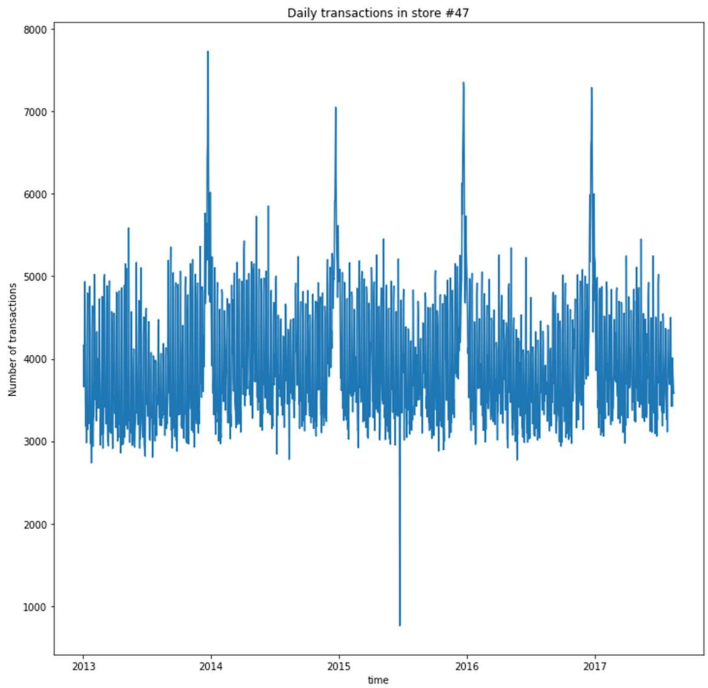
The data provided is a table with the following variables/characteristics: date, warehouse number, article number, sales volume, promotion. While the store ID and article ID are integers, the promotion is a Boolean and the sales volume is a float (integers for discrete articles, float for volume/weight). The company also offers other data sets, such as a list of stores with their locations, a time series of daily transactions per store, a list of holidays and events, a list of products by category and the price of oil, to which much of the Ecuadorian economy is said to be linked. These are additional tools to simplify and/or improve forecasting, and some other external data could also be used in this regard.

	date	store_nbr	transactions	city	state	type	cluster	year	week	day	dayoff
0	2013-01-01	25	770	Salinas	Santa Elena	D	1	2013	1	2	True
1	2013-01-02	1	2111	Quito	Pichincha	D	13	2013	1	3	False
2	2013-01-02	2	2358	Quito	Pichincha	D	13	2013	1	3	False

The data can be viewed as N time series, one per combination (store, material). Many of these time series are most likely correlated with each other and a dimensional reduction will be welcome here.

[7] Available at this address https://www.kaggle.com/c/favorita-grocery-sales-forecasting

The main task is therefore to make forecasts. For the sake of simplicity, and since this is the first time series, this core will focus on the daily transaction data of a single store instead of the sales data of all stores.

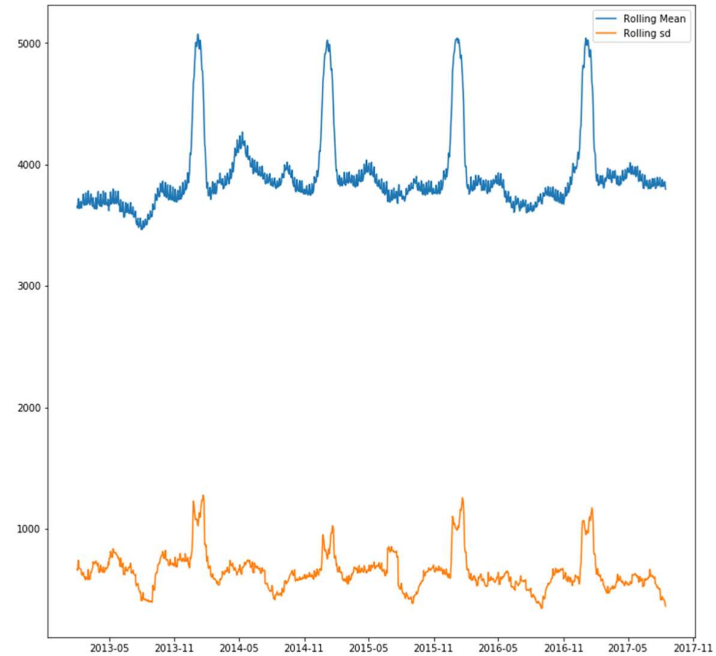


Data Profile

We can easily identify a peak around Christmas, a very low volume day in mid-2015 and a stable behavior in between. We were expecting a volume increase around Christmas, but what happened in mid-2015? It turns out that this corresponds to a gigantic national event in Ecuador, which will most likely have repercussions on other stores as well[8].

We can also see that outside of this holiday period, transactions are very variable, with volume varying between 3,000 and 5,000. The moving average and the standard deviation with a window of one month can be verified by means of the Dickey-Fuller test, which has the null hypothesis that the time series is not stationary. The result being p-value = 4.982766e-10, we can safely reject it and conclude that our time series is indeed stationary.

[8] https://en.wikipedia.org/wiki/2015_Ecuadorian_protests#24_June



Research tracks

Now that we have a clearer picture of the type and profile of data, the first step in this work has been to conceptualize what a stock management program would look like. Should we have a program that proposes a totally different and counter-intuitive analysis of the data or should we approach human reasoning by a succession of small algorithms that would allow the program to propose a sequence of small solutions, which could be accepted, modified or abandoned by an expert who would supervise the program.

The first hypothesis seemed to me uninteresting to deal with in the context of this thesis. Indeed, if on the basis of raw data entered in a black box, the algorithm proposed relevant solutions, the explanation of such results would be impossible for me to give from a logical point of view. Only the explanation of the mathematical methods behind the program would eventually have been within my reach.

The second hypothesis seemed much more interesting to me. It allows us to give an objective answer to ideas that we might have about the functioning of the economy, to refute received ideas and false relations of cause and effect, and to give reason to human intuition by proposing a succession of steps that are not understandable to humans. It also offers the possibility of opening up new avenues and abandoning some of them without forgetting everything.

Thus, I tried to divide the problem of sales predictions into sub-problems.

The first problem that came to my mind was to know the algorithm capable of determining the sequence of numbers (i.e. sales) in order to determine future sales. Indeed, the cyclical side of the seasons is a factor to be considered in the training of the model and algorithms had to exist for that. This is a regression problem.

A second problem consists in dealing with the correlation that can be found in the products and the influence of the latter was no longer respected. Typically, one can imagine two products that are often sold together (typically the pair of shoes with the spare laces that go with them) and wonder whether observing a difference in trend would not be a warning message indicating that one is in a non-comparable situation. This problem would therefore be more of a classification issue.

Recurrent neural networks

Introduction

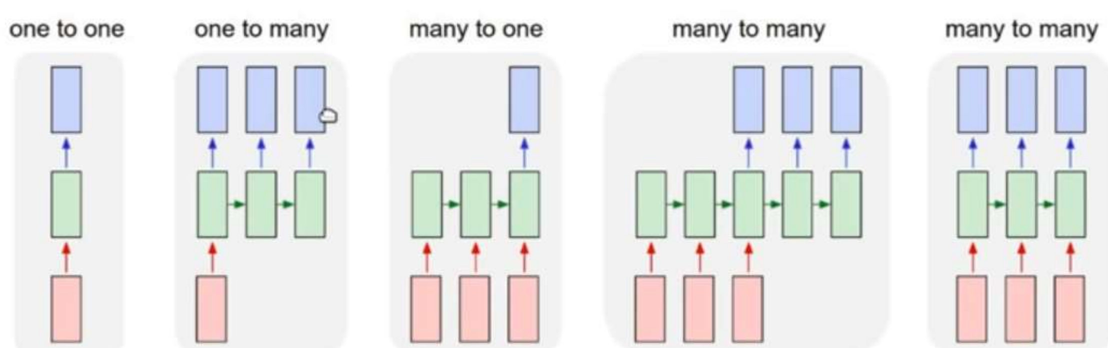
Recurrent neural network algorithms are reasoning used extensively in Deep Learning and are very common in everyday life, such as when we talk, and a device tries to understand what we are saying.

Recurrent neural networks are algorithms that process sequences, unlike other algorithms such as the classical neural network that takes an X vector of data and returns a Y vector, based solely on the X vector. They thus make it possible to manage inputs of varying sizes (the voice). NNRs can consider sequences of vectors, more commonly known as series.

To continue in the example of speech recognition, a sentence is composed of words that are made up of syllables. The RNN will therefore guess one syllable and then another syllable to form words and reconstruct a sentence. Obviously, some syllables come more frequently than others after a first syllable, hence the reason for being able to consider the previous syllables to recognize the following syllables.

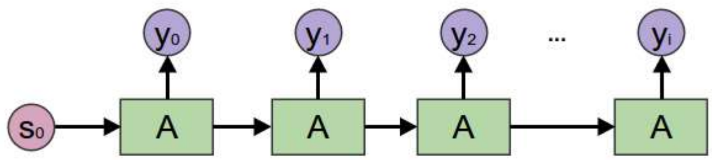
The images and explanations of the reasoning are based on a video explaining what an RNN is. The understanding as well as the handling of these systems is an integral part of the course, allowing me to better understand the tools commonly used for similar machine learning problems.

Types

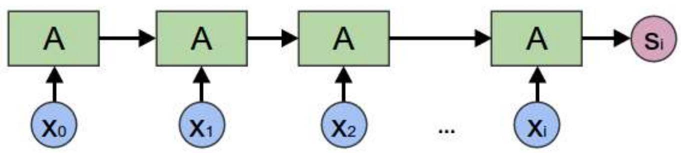


Thus, several cases are available to us

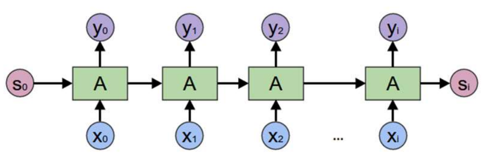
1) The one to one: this is the classical neural network where for a single vector X we have only one output Y.
2) The one to many: for a single X vector as input we have a series of Y vectors as output. Typically when we write a first word in an SMS and the phone tries to guess the following words.

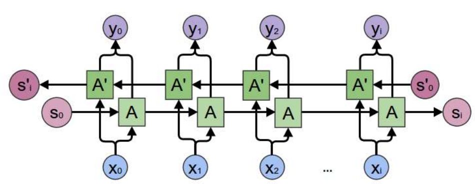


3) The many to one: entering a sentence, is what the program understands if it is positive or negative for example. As part of stock prediction.



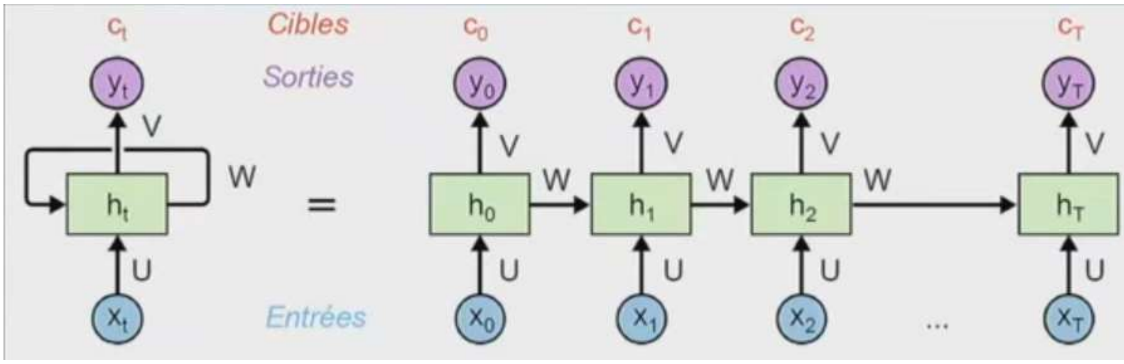
4) The many to many: the first case is typically illustrated when a program tries to translate a sentence, it will wait until it has the whole sentence to translate. In the second case it is for example when we want to transcribe what is said orally (typically the subtitle generation algorithm in videos).





Operation

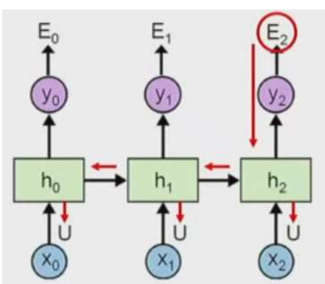
A recursive neural network will therefore be composed as follows: an input sequence (xT), an output sequence (yT), and an internal state (hT).



The global error is measured by the sum of the errors (difference between the target ci and the output value yi). Thus, to train the model, we arrive at a classical back propagation through time to adapt the parameters.

Let's consider this 3-step system. The calculation of the global error is expressed as the sum of the errors E0, E1 and E2 which are the differences between the expected values and the values obtained at each step. In order to train this system, we must therefore minimize the error.

The reasoning is like the classical neural network. In order to determine the gradient on U for example, we will take the partial derivative with respect to each of the errors in order to isolate all the parameters h, variables which modify the behavior. As an example, here is the calculation for E2 which must therefore be added to E1 and E0 to determine the partial derivative between the total error and the matrix U.



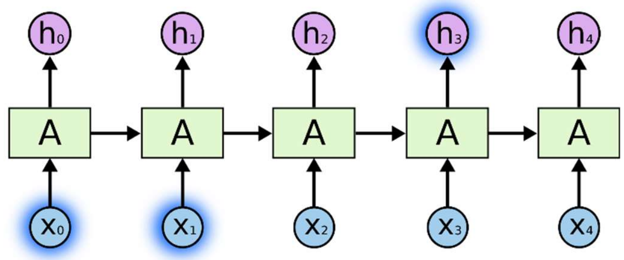
$$\frac{\partial E_2}{\partial U} = \frac{\partial E_2}{\partial h_2}\left(x_2^T + \frac{\partial h_2}{\partial h_1}\left(x_1^T + \frac{\partial h_2}{\partial h_1}x_0^T\right)\right)$$

We can see that to calculate one of the terms of a 3-step recursive neural network, the calculations are already complex and the optimization not so simple.

Problem and origin of LSTM style algorithms

The NRNs thus give the idea that they might be able to link previous information to the current task, such as the use of previous video images that might shed light on the understanding of the current image. If RNNs could do this, they would be extremely useful. But can they? That depends.

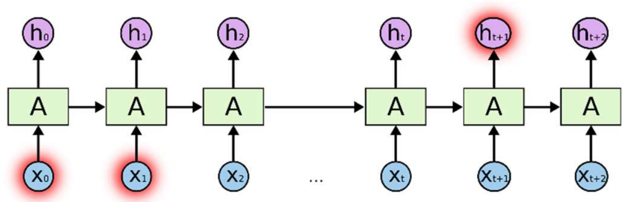
Sometimes it is enough to look at recent information to complete the current task. For example, consider a linguistic model that tries to predict the next word based on previous ones. If we are trying to predict the last word in "the clouds are in the sky", we don't need any additional context - it is obvious that the next word will be "sky". In such cases, when the gap between the relevant information and where it is needed is small, RNNs can learn to use past information.



But there are also cases where we need more context. Think about trying to predict the last word of the text "I grew up in France... I speak fluent French." Recent information suggests that the next word is probably the name of a language, but if we want to narrow down the list of languages, we need the context of France, going back further in time. It is quite possible that the gap between the relevant information and the point at which it is needed could become very large.

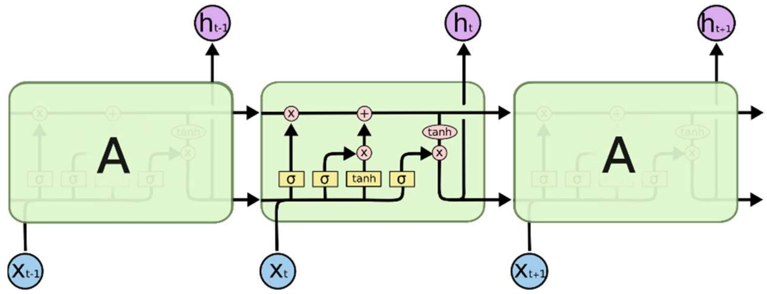
In the hope of being able to predict stocks, is the information of the last few days enough to determine future stocks? The worst case must be considered.

Unfortunately, as this gap widens, NRNs become unable to learn how to relate information to each other.

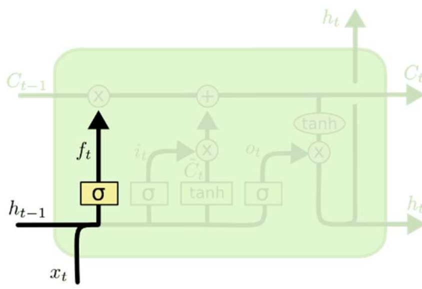


LSTM

Long Short-Term Memory Networks (LSTM) - are a special type of LSTN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997) and have been refined and popularized by many people in subsequent work. They work extremely well for a wide variety of problems and are now widely used. MSTLs are explicitly designed to avoid the problem of long-term addiction. Remembering information for long periods of time is virtually their default behavior. All recurrent neural networks are in the form of a chain of repetitive neural network modules. In standard RNNs, this repetitive module will have a very simple structure, a single layout. The LSTM, on the other hand, has four.

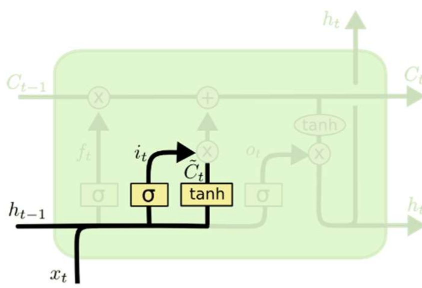


Step 1: Simplification



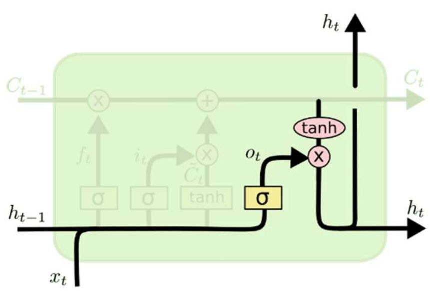
The objective here is to be able to determine which information is useful and which is no longer to be considered. Based on the memory at the previous instant and the data retrieved at instant t, this step decides for each component of the vector C whether the value should be reset or not.

Step 2: Adjustment



After removing the superfluous information, a decision must be made as to what information is to be retained. Based on a sigmoid and the candidate values in CT, the values are added to the data that have passed step one.

Step 3: filter



The idea behind this step is to transmit the state of the cell for the next step and thus ensure continuity in learning (that the algorithm does not contradict itself from one step to the next).

Case study in the case of stock management

The idea of incorporating LSTM into my algorithm came up at several points.

First, I thought that since it was a sequence processing tool, it would be able to process a sequence of sales to determine future sales. Unfortunately, it doesn't work. Indeed, LSTM is still an RNN which remains a neural network. These algorithms deal with a classification problem and not a regression problem. It is therefore impossible to classify a future because it can have an infinite number of facets, just as training would require knowing all possible pasts, which is also impossible.

A second case of use, and one that I think would make much more sense, would be to determine, precisely based on a sequence of sales, which are comparable cases. In the context of the Van Bommel analysis, it would have made sense to base a classification of the pairs of shoes on their history. In this way, when the season ends and a new collection arrives, it can be assumed that the new successful model will follow the same curve as its old model. Unfortunately, this step could not be implemented because the data was not accessible, and no other database offered a similar data set.

Time series

Introduction

There are two possible approaches to dealing with time series. On the one hand, the data can be analyzed as a function of time. Then the objective will be to determine which function best represents the curve (by the least squares method or other iterative methods). On the other hand, the data can be treated as a function of the values preceding it. This second step represents the set of ARIMA models.

Model ARIMA

The ARIMA[9] model is used on this time series to see to what extent we can predict future events based on past events. ARIMA stands for AutoRegressive Integrated Moving Average. Further explanation

ARIMA models have three parameters, corresponding to its AR, I and MA components:

- The number of autoregressive terms (AR, p): The AR terms are only delaying of the dependent variable.
- The number of differences (I, d): This is the number of non-seasonal differences.
- The number of moving average terms (MA, q): The MA terms are lagged forecast errors in the forecast equation.

Differentiation

The estimation of ARIMA models assumes working on a stationary series, in other words a series where the mean as well as the variance remains constant over time. Thus, the d parameter of the model

[9] https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

allows to replace the original series by the series of adjacent differences. If the data are stationary, we can already define the parameter d as null.

Autoregression

Parameter p is used to control autoregression. Autoregressive means that the model expresses the data (or difference in the case of a delayed model) by a linear combination of the previous values cumulated with a random component.

Moving average

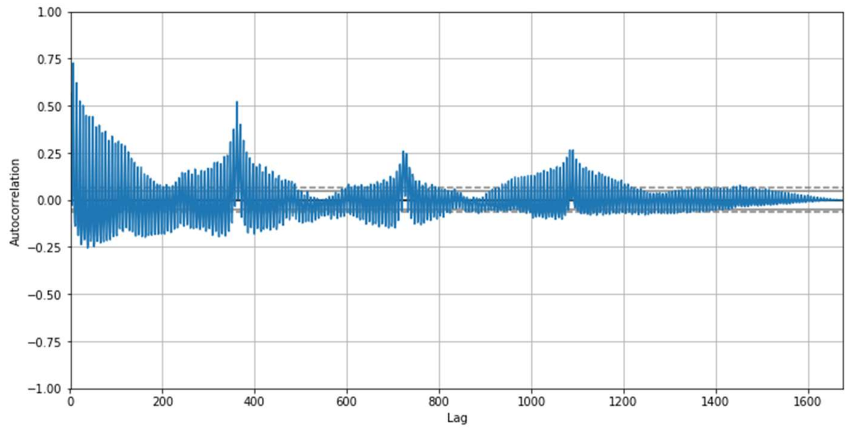
This last parameter suggests that the series fluctuates around a mean value and that the best estimate is represented by an average of previous observations (data smoothing). Thus the estimate would be equal to this mean value plus a weighted sum of the errors due to the previous values.

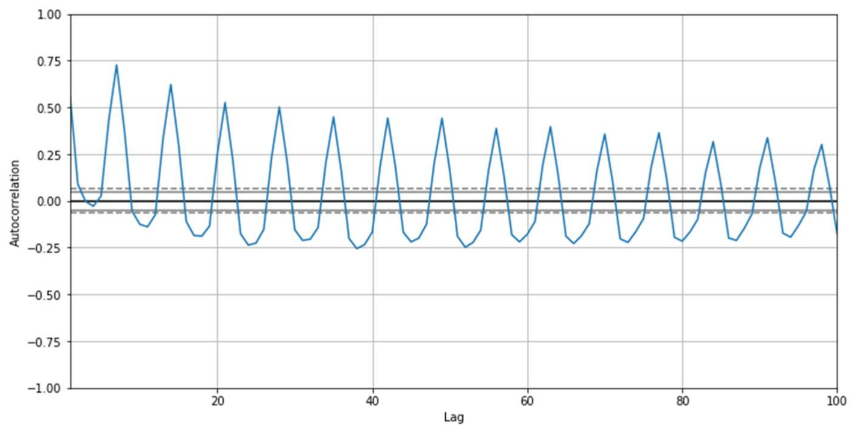
Practical significance for the ARIMA model

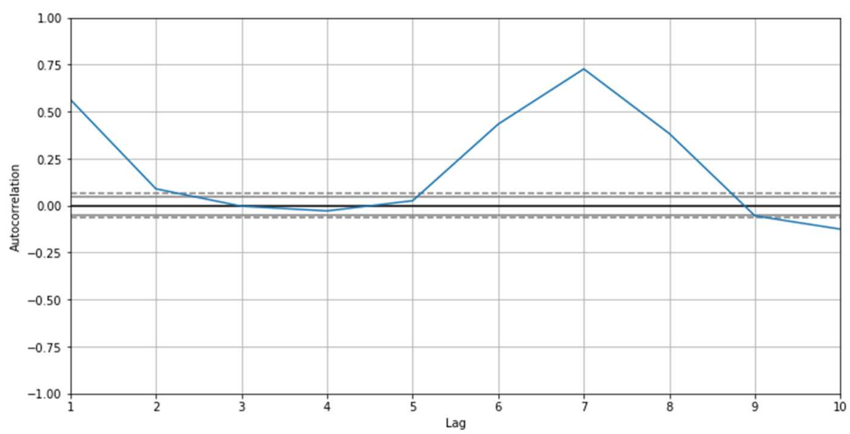
ARIMA(0,0,0) suggests an undifferentiated white noise process, i.e. random fluctuations around a reference value. This reference value can be considered as a stable characteristic of the system under study (personality trait, memory, stabilized capacity, etc.). The moving average process, on the other hand, suggests that the reference value changes from one measurement to the next, specifically that the reference value is a function of the previous reference value and the error of the previous measurement. Finally, an auto-regressive process suggests that the phenomenon under study is not determined by a reference value. It is the previous performance (or previous performances) that fully determines the current performance.

Practical case of the thesis

We will set d to zero and will not use the integrated part of the model. Let's start by checking the autocorrelation of our time series to better understand it and to determine what the parameters of our model should be.



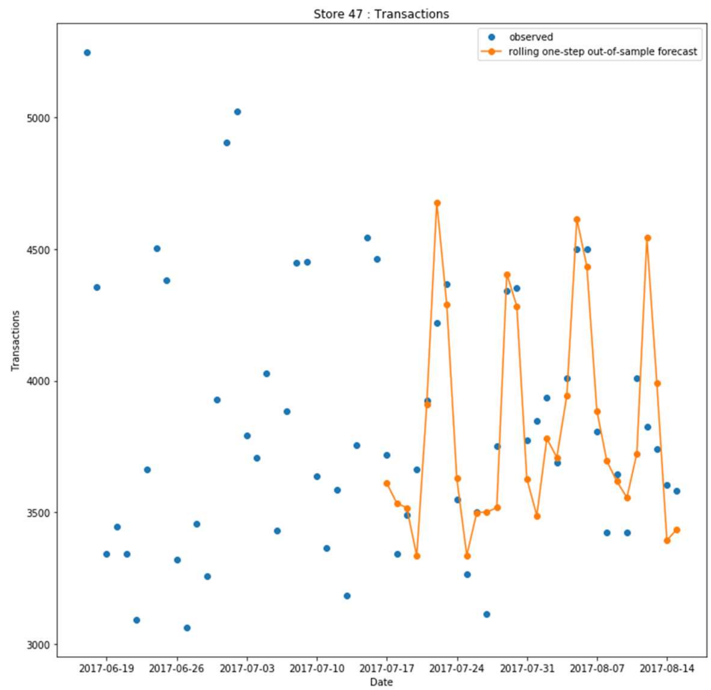




The highest autocorrelation peaks occur approximately every 365 days and oscillate with a frequency of about 7 days. This is consistent with the intuitive idea of the distribution of annual and weekly sales.

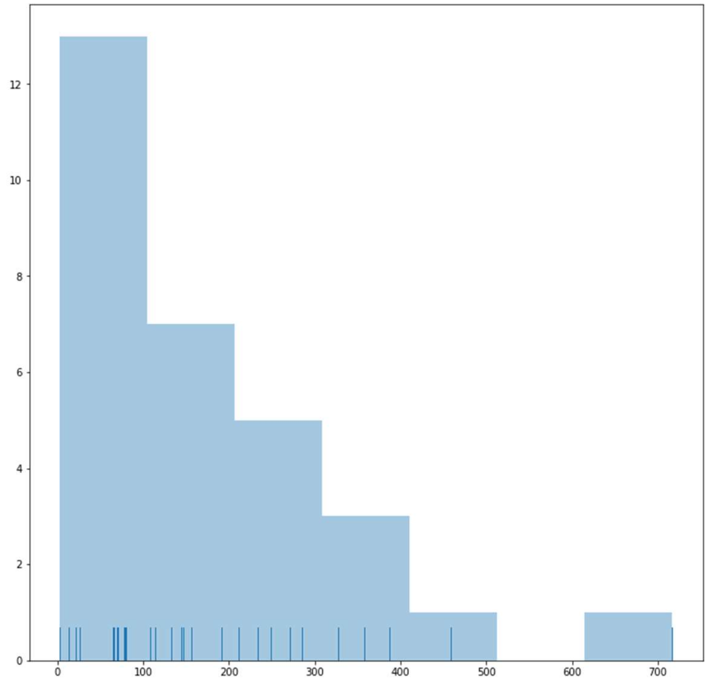
From the last diagram, it appears that the autocorrelation is significant (above the dashed line) for a period of up to 2 days. All references then suggest using 2 for the parameter p.

However, the *stats.model* library contains a tool for coefficient selection*: arma_order_select_ic().* It allows a grid search with the parameters p and q. By choosing 10 as max parameter and the Bayesian information criterion[10] for the estimator, the proposed choice is *(5,0,5)* for the three parameters of the ARIMA model. Using these parameters, we obtain this graph:



The results look promising. In fact, 66% percent of the estimates are more than 95% accurate but this means that already 33% of the results are approximated with difficulties, whereas the objective was simply to estimate the number of transactions.

[10] https://fr.wikipedia.org/wiki/Crit%C3%A8re_d%27information_bay%C3%A9sien



Model improvement and conclusion

So, the question becomes: how to make the algorithm more efficient?

The answers are multiple and require further study each one:

1. The model is not good: this first answer is quite plausible because we don't know the mathematical laws that govern the rules of the market. By making estimates, we try to approximate the information gathered in the past by a simpler mathematical model, but we have no idea whether this model is correct or not, since it is based only on past data without being able to adapt to future data. An ideal model should be able to readjust itself naturally based on the data it collects daily and should be constantly learning. In fact, to date, we can only re-train the model once the measurements become more and more erroneous.
2. Lack of information: once again, the answer is totally plausible since even today an event like COVID-19, which is not considered by the algorithms, has major impacts on the world economy. But on a smaller scale, the departure or arrival of a competing trader, or the promotion of the product has a much greater impact on sales compared to the trend of recent years. These problems are all the greater since it is not possible to measure the consequences to date.
3. Observables are seen at the wrong scale. This answer is more than intuitive, although the answer is much less so. It is known that it is by looking at a large scale that the relative error is less because local fluctuations compensate each other (principle of the law of large numbers

in terms of statistics, therefore we base ourselves on several years of learning). However, looking at all the stores in order to determine the sales behavior of a business in a small village seems as contradictory as it is aberrant.

Different approaches can be considered, all of which depend on the type of data being processed.

1. Analyzing the sales trends of the different outlets by comparing their sales profile over the last few years. An immediate subsidiary question would be how to compare these sales outlets? Annual turnover, monthly daily turnover? Or rather percentage growth of the store's turnover? A lot of factors can be considered, which makes it even more difficult to train the model. In our dataset, a classification has already been made and yet, the very sales profile vis-à-vis this classification made by the company differs within two stores of the same group.
2. Addition of external factors such as the weather or at least their forecast, gas prices, atypical holidays, etc. These additional data, as we know, are subject to be a factor in the evolution of sales and can certainly refine the results. But can the training cost of the model really be offset by the few percentages of precision we gain?

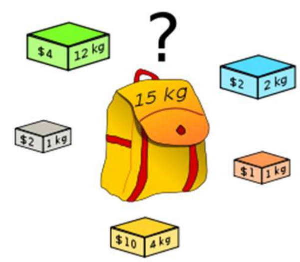
In the context of the sale of fashion clothing and shoes, an essential element demonstrates the difficulty of comparing the sale of a product of this year with its previous years. The fashion effect as well as the old and new collections. Every customer is different, and the position of the product in the store has its role. But one thing is for sure, when buying a garment, what the person has seen and heard outside the store will play an important role, as will the change in the layout of the products in the store. Some establishments prefer to adapt the layout of their products in their stores rather than returning stock, and this is to their credit since transport is at a cost. But it must be realized that this then creates an error in the predictions.

Convoy management

Introduction

Convoy management is the second part of what the algorithm should be capable of. Once speculations are made on future sales, the objective is to take advantage of this information to adjust the stock judiciously. This includes the balancing (overstock and understock) but also the planning and organization of convoys. How often should stores be restocked? In this chapter we will see the difficulty of being able to solve this problem based on the experience I gained while working as a student in a decathlon store.

Algorithms Knapsack



The knapsack problem, which as been studied for more than a century is a problem in combinatorial optimization:

Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

This algorithm seems essential for automated replenishment. But although its statement is clear and resolution paths explored for the most part, the difficulty is actually quite different.

As we know, the goal is to maximize profit and minimize costs and losses. With this in mind, we may be interested in using the backpack algorithm to, for example, prioritize orders (choose which orders to process first). We know that in the statement we talk about value and weight. How to determine these two values in the context of optimized replenishment?

Let's take the example of weight. Is it only a question of the space required and the distance to be covered for each order? Or should we only be interested in the journey time because we assume the trucks are big enough?

The difficulty of this algorithm is therefore much more in its enunciation than in its resolution. Moreover, many algorithms propose an approximate solution for a time saving. The difficulty comes down to modeling what each decision taken by the algorithm costs and pays back.

Simulation vs reality

Sub-stock

Let us first process the sub-stock and thus the procurement from the warehouse to the storage location. Usually, when a warehouse places an order and requests to be restocked with a product, the order takes some time to be effective. Depending on the store, this delay can be longer or shorter.

Typically, at Decathlon, there are two types of orders: the first is the order from the warehouse, and the second is the order from another store, which is more commonly referred to in the jargon as "intermag".

The order from the warehouse is the most logical and stable, as it is usually processed within a few days. Intermag usually takes longer, because the transport is always from the warehouse to the warehouse and only then from the warehouse to the second warehouse.

The store where I worked (Decathlon Evere) received 2 supplies per day, one in the morning and one in the early afternoon. These orders only processed orders from previous days. And depending on the availability of what was available in the Willebroek warehouse, these orders were closed within a few days or months.

When a customer arrived in the store and wanted to buy a product that was not available, we had several scenarios:

1. The order was already in progress and we just had to wait for the right supplies to fill the shelf. In this case, the customer usually waited and came back either in the afternoon or the next day. A marketing gesture was then to take his number and call him when the product was available.
2. There were no orders in progress. In this case, we could check the stock of the other stores and the warehouse and choose to either intermag or place an order with the warehouse depending on whether the product was available or not. Usually when the product was not available, and intermag was necessary, the salesmen preferred to first suggest to the customer to go to the second store of the chain in order to obtain the product directly.
3. An error in the recorded stock. In a store such as the Decathlon store, it is common that the stock indicated on our control devices is not exact with the stock found on the shelf. There are many reasons for this:

- The store offers the product in several places in the store.
- The store had an overstock a few days ago, which caused it to have to put the surplus in reserve but did not reuse it once the shelf had run out.
- The product is still in the store but not yet at the checkout (left in the wrong place or in another customer's cart, returns, etc.).
- Unloading is in progress (between the moment the product leaves the truck and the moment it is put in the shelf, a period that can be relatively long).

Thus, these few elements show that despite regular orders, stock errors can occur in the warehouse and that it is necessary to have a margin between the required stock and the proposed stock. This margin depends of course on many factors such as the size of the store, its turnover, the ethics of the store, its policy in case of overstock etc.

We also note that all this information is very random and difficult to control by an algorithm that is intended to be constant and optimized.

Finally, we note that there is also a human limit that limits the optimization of the algorithm in terms of supply. A too important supply generates the unavailability of the seller for the customer, but also the risk of dissatisfaction of the latter who wishes to acquire his product that he saw available online, but which is still in the supply boxes. Too low a supply would require more frequent refueling, which would cost the company a lot in terms of fuel.

Overstock

Overstocking is also a difficult subject to consider by the algorithm. Between the shortage that can be observed in other stores that would push intermag, or simply overly optimistic expectations regarding sales, Decathlon Evere has the flexibility to adjust its product layout to maximize customer accessibility. For example, it is common that in the event of a slight overstock, the department decides to increase the space allocated to the product. If the overstock becomes more important, the department may consider arranging the product in TG (gondola head, visible side of the shelf in the aisles of the store). This is usually a hot spot that is very likely to lead the customer to buy a product for which he did not come. If this is not enough, the store may store some of it in its stock, but the store will always think twice before returning stock to the warehouse as this is the most expensive. As a result, it is quickly understood that a store will completely change the expected behavior of the product's sales profile in order to accelerate sales of that product. For an algorithm, this factor is very difficult to deal with because it depends on the subjectivity of the salesperson and his decision making.

Conclusion

We can see that around a balance, the store is able to work efficiently. Although the customer may sometimes find himself in a situation where his product is not available, the salesperson will nevertheless try to avoid any loss of income for his company by redirecting the person concerned to a similar product, to another store, or by trying to make him patient. Usually the balance is made to the human feeling and capacities. We can see that a store with a large stock will spend more energy to keep this stock in order, contrary to a smaller store that will often reach breakdowns in the products it offers. So how do you determine the balance? Should we be satisfied with a few products that we offer in large quantities? Is it necessary to offer a multitude of products in only a few copies? These questions call for testing the different options, hoping that the cases will be under comparable conditions. Obviously, for a young student about to graduate, it is difficult to obtain such confidence from the company.

Acknowledgements

First, I would like to thank Patrick Tack for letting me once again benefit from the trust of Avanade and the instruments put at my disposal for this thesis. Of course, I include all the team he supervises and who were able to help me in this work.

My thoughts also turn to Sébastien Combéfis, the ECAM contact person for this TFE. He will have been available in a flexible way to answer my concerns and questions. He will also have been able to help me manage this end of TFE in this atypical period of confinement.

Finally, I would like to thank ECAM for these academic years, which have enabled me to acquire not only knowledge but also a methodology for tackling this subject.

Conclusion

After all these months and all these hours of research and learning, I am still happy with the work done even though I admit that I am still hungry.

First of all, even if the adoption of the Azure environment was indeed brief when the denial of access to the Van Bommel company's data was announced, the researches allowed me to understand the necessary steps from the incorporation of the data in Azure to the export of the decisions made by the algorithm. The research could not indeed be put into practice but gave me a feeling that I had already felt when I had done my internship with them, I am able to tame Microsoft tools and its environment. Indeed, the steps to take were clearer, I already knew partially the different popular Microsoft tools, I was therefore able to simply focus my research on the communication between these tools and thus find the steps to follow more quickly.

Secondly, even if in my opinion the stock prediction algorithm is not finished, my reasoning as an engineer was not meaningless because for every question I asked myself, I found an answer on the Internet, the disadvantage being probably that I could not find it right away. The idea of using LSTM is indeed evoked in scientific articles, the ARIMA model also seems to be at the forefront of sequence prediction. The problem of autocorrelation as well as of the new collection vis-à-vis the old ones are parameters that I had directly highlighted and which to date have no solution.

Furthermore, the convoy management, and more particularly the algorithm with multiple backpacks is typically an algorithm that left little room for imagination as the problem is so well known and mastered. Whether for the STIB, or for UBER convoys, this problem is found everywhere. However, in the context of stock replenishment, this problem cannot be expressed in this way because it only considers a minority of factors.

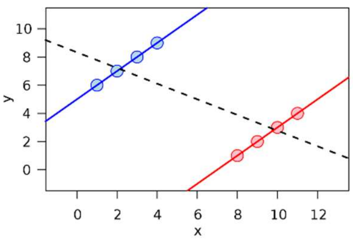
Finally, let's imagine the case where we manage to take into account all the factors and that we manage to build a model on the basis of these factors, I wanted to keep this last paragraph as an illustration of my conclusion. It deals with the difficulties and paradoxes that learning algorithms can face in modelling. These paradoxes are the very image of the need for human control behind any program.

Through this thesis, I came up with many ideas, the hope of even creating a more efficient management than the human one, but I gained in maturity by reading a lot of articles, examples of algorithms, which show the random side of the possibility to find a model and to check if it is good or not. I concluded that this could not be done in a single year and that it would take years of research. It is not for nothing that Google and Kaggle offer competitions in this kind of reasoning.

In the depths of the difficulties of learning machines: paradoxes

Like any form of data-based knowledge construction, machine learning models are not free of cognitive paradoxes. On the contrary, because machine learning attempts to infer models hidden in training datasets and to validate their knowledge against a specific environment, they are constantly vulnerable to paradoxical conclusions. Here are some of the more notorious paradoxes that surface in machine learning solutions.

The Simpson's Paradox



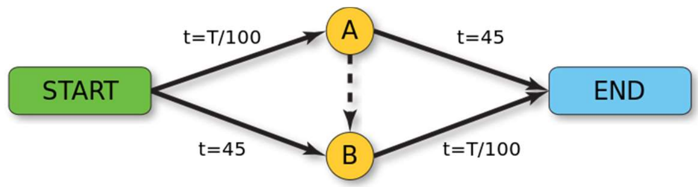
Named after the British mathematician Edward Simpson, Simpson's Paradox describes a phenomenon in which a very apparent trend in several groups of data dissipates when the data from these groups are combined. A real case of this paradox occurred in 1973. Admission rates were studied in the graduate schools of the University of Berkeley. The university was sued by women for the gender gap in admissions. The results of the survey were as follows: When each school was examined separately (law, medicine, engineering, etc.), women were admitted at a higher rate than men! However, the average suggests that men are admitted at a much higher rate than women. How is this possible?

The explanation for the previous use case is that a simple average does not consider the relevance of a specific group within the data set. In this specific example, women applied in large numbers to schools with low admission rates: such as law and medicine. These schools admitted less than 10% of the students. As a result, the percentage of women accepted was very low. Men, on the other hand, tended to apply in greater numbers to schools with high admission rates: such as engineering, where admission rates are around 50%. As a result, the percentage of men accepted was very high.

In the context of machine learning, many unsupervised learning algorithms infer patterns different training datasets that result in contradictions when combined across the board.

The Braess's Paradox

This paradox was proposed in 1968 by the German mathematician Dietrich Braes. Using an example of congested road networks, Braes explained that, counterintuitively, adding a road to a road network could potentially impede its flow (e.g. the travel time of each driver); similarly, closing roads could potentially improve travel times. Braess' reasoning is since in a Nash equilibrium game, drivers have no incentive to change their routes. In terms of game theory, an individual has nothing to gain from applying new strategies if others stick to the same ones. In the case of drivers, a strategy is a route taken. In the case of the Braess paradox, drivers will continue to change routes until they reach Nash equilibrium, despite the reduction in overall performance. Thus, counterintuitively, road closures could reduce congestion.



The Braess paradox is highly relevant in autonomous learning and multi-agent reinforcement scenarios in which models must reward agents based on specific decisions in unknown environments.

Moravec's paradox

Hans Moravec can be considered one of the greatest AI thinkers of the last decades. In the 1980s, Moravec made a counter-intuitive proposal on how AI models acquire knowledge. Moravec's paradox states that, contrary to popular belief, high-level reasoning requires less computation than low-level unconscious cognition. This is an empirical observation that runs counter to the notion that greater computational capacity leads to smarter systems.

A simpler way of formulating Moravec's paradox is that AI models can perform incredibly complex statistical and data inference tasks that are impossible for humans. However, many tasks that are insignificant to humans, such as grasping an object, require expensive AI models. As Moravec writes, "it is relatively easy to get computers to perform at an adult level in intelligence tests or checkers games, and difficult, if not impossible, to give them the perceptual and mobility skills of a one-year-old child.

From the point of view of machine learning, Moravec's paradox is very applicable in those aspects of transfer learning that aim to generalize knowledge through different models of machine learning. Furthermore, Moravec's paradox tells us that some of the best applications of artificial intelligence will be the result of a combination of humans and algorithms.

The accuracy paradox

Directly related to machine learning, the accuracy paradox indicates that, against all expectations, accuracy is not always a good measure for ranking the efficiency of predictive models. How can this be confusing? The accuracy paradox has its roots in unbalanced training data sets. For example, in a data set where Category A incidence is dominant, occurring in 99% of the cases, then predicting that each Category A case will have 99% accuracy is completely misleading.

A simpler way to understand the accuracy paradox is to find the balance between accuracy and recall in machine learning models. In machine learning algorithms, accuracy is often defined as a measure of the fraction of your predictions for the positive class that is valid. It is formulated as (True Positive / True Positive + False Positive). In addition, the recall measure measures how often your predictions capture the positive class. It is formulated as (True Positive / True Positive + False Negative).

In many machine learning models, the balance between accuracy and recall gives a better measure of accuracy. For example, in the case of a fraud detection algorithm, recall is a more important measure. It is obviously important to detect all possible frauds, even if it means that the authorities might need to go through a few false positives. On the other hand, if the algorithm is created for sentiment analysis and all you need is a high-level idea of the emotions indicated in the tweets, then accuracy is the way to go.

The paradox of learning ability

Saving for the end the most controversial, it is a very recent paradox that was published in a research paper earlier this year. This paradox links the ability of an automatic learning model to one of the most controversial mathematical theories: Gödel's incompleteness theorem.

Kurt Gödel is one of the most brilliant mathematicians of all time and has pushed the boundaries of philosophy, physics and mathematics like some of his predecessors. In 1931, Gödel published his two incompleteness theorems, which essentially say that certain statements can be proven neither true nor false using standard mathematical language. In other words, mathematics is an insufficient language for understanding certain aspects of the universe. These theorems are now known as the Gödel continuum hypothesis.

In a recent work, AI researchers at the Israel Institute of Technology have linked the Gödel Continuum Hypothesis to the learning ability of a machine learning model. In a paradoxical statement that challenges all common wisdom, the researchers define the notion of learning limbo. Essentially, the researchers go on to show that if the continuum hypothesis is true, a small sample is enough to extrapolate. But if it is false, no finite sample can ever be enough. Thus, they show that the learning ability problem is equivalent to the continuum hypothesis. Therefore, the learning ability problem is also in a state of limbo that can only be solved by choosing the axiomatic universe.

In simple terms, the mathematical evidence of the study shows that AI problems are subject to Gödel's continuum hypothesis, which means that many problems could be effectively unsolvable by AI. Although this paradox has very few real-world applications for AI problems today, it will be crucial for the evolution of the field soon.

Paradoxes are ubiquitous in real-world machine learning problems. It can be argued that algorithms, having no notion of common sense, could be immune to statistical paradoxes. However, since most machine learning problems require human analysis and intervention and are based on man-made data sets, we will live in a world of paradoxes for some time.

Sources

- https://mc.ai/moravecs-paradox-implies-that-agi-is-closer-than-we-think/
- Knapsack algorithm, https://www.youtube.com/watch?v=xOlhR_2QCXY
- Databricks documentation, https://docs.databricks.com/notebooks/notebook-workflows.html#example
- Kaggles notebooks and datasets, https://www.kaggle.com/
- RRN, C. Laurent, https://www.youtube.com/watch?v=dOpgDv88UOo
- Colah Blog, C. Colah, https://colah.github.io/
- Five machine Learning paradoxe, J. Rodriguez https://towardsdatascience.com/five-machine-learning-paradoxes-that-will-change-the-way-you-think-about-data-e100be5620d7
- Microsoft documentation, https://docs.microsoft.com/en-us/azure/?product=featured
- Thibault Neveu youtube, https://www.youtube.com/watch?v=3xgYxrNyE54
- Wikipedia for its definitions
- ECAM courses, especially computer courses but also automatic
- Openclassroom for its AI courses about machine learnings
- Pluralsight for its AI solution in Azure