Autonomy is all you need

Romain Avouac, Frédéric Comte and Thomas Faria

Abstract Abstract here

1 Introduction

In recent years, the European Statistical System (ESS) has committed to leverage non-traditional data sources in order to improve the process of statistical production, an evolution that is encapsulated by the concept of Trusted Smart Statistics [20]. This dynamic is accompanied by innovations in the statistical processes, so as to be able to take advantage of the great potential of these new sources (greater timeliness, increased spatio-temporal resolution, etc.), but also to cope with their complexity or imperfections. At the forefront of these innovations are machine-learning methods and their promising uses in the coding and classification fields, data editing and imputation [13]. The multiple challenges faced by statistical institutes because of this evolution are addressed in the Bucharest Memorandum on Official Statistics in a Datafied Society (Trusted Smart Statistics), which predicts that "the variety of new data sources, computational paradigms and tools will require amendments to the statistical business architecture, processes, production models, IT infrastructures, methodological and quality frameworks, and the corresponding governance structures", and consequently invites the ESS to assess the required adaptations and prioritize them [7].

In line with these recommendations, much work has been done in the context of successive projects at the European level in order to operationalize the use of non-traditional data sources in the production of official statistics. Within the scope of the ESSnet Big Data II project (2018-2020), National Statistical Offices (NSOs) have

Romain Avouac
Insee, 88 avenue François Verdier, Montrouge, e-mail: romain.avouac@insee.fr

Thomas Faria
Insee, 88 avenue François Verdier, Montrouge, e-mail: thomas.faria@insee.fr

been working across a wide range of themes (online job vacancies, smart energy, tracking ships, etc.) in order to put together the building blocks for using these sources in actual production processes and identify their limitations [9]. However, while a substantial amount of work has been devoted to developing methodological frameworks [6, 22], quality guidelines [15] as well as devising business architectures that make third-party data acquisition more secure [19], not much has been said about the IT infrastructures and skills needed to properly deal with these new objects.

Big data sources, which are at the heart of Trusted Smart Statistics, have characteristics that, due to their volume, their velocity (speed of creation or renewal) or their variety (structured but also unstructured data, such as text and images), make them particularly complex to process. Besides, the "skills and competencies to automate, analyse, and optimize such complex systems are often not part of the traditional skill set of most National Statistical Offices" [3]. Not incidentally, an increasing number of public statisticians trained as data scientists have joined NSOs in recent years. Within its multiple meanings, the term “data scientist” reflects the increased involvement of statisticians in the IT development and orchestration of their data processing operations, beyond merely the design or validation phases [4]. However, the ability of these new data professionals to derive value from big data sources and/or machine learning methods is limited by several challenges.

A first challenge is related to the lack of proper IT infrastructures to tackle the new data sources that NSOs now have access to as well as the accompanying need for new statistical methods. For instance, big data sources require huge storage capacities and often rely on distributed computing frameworks to be processed, which generally cannot be provided by traditional IT infrastructures [17]. Similarly, the adoption of new statistical methods based on machine learning algorithms often require IT capacities (in particular, GPUs - graphical processing units) to massively parallelize computations [21].

Another major challenge is related to the difficulty of transitioning from innovative experiments to production-ready solutions. Even when statisticians have access to development environments in which they can readily experiment, the step towards putting the application or model in production is generally very large. Such examples highlight the need to make statisticians more autonomous regarding the orchestration of their processings as well as fostering a more direct collaboration between teams, as advocated by DevOps and DataOps approaches.

A third challenge is to foster reproducibility in official statistics production. This quality criterion involves devising processing solutions that can produce reproducible statistics on the one hand, and that can be shared with peers on the other hand.

- Final challenge : encourage and faciliate collaboration - Against that background, we argue that common theme : fostering autonomy - ref innovation plateformes blabla - choix technologiques qui favorisent l’autonomie et la scalabilité - make cloud resources easily available - retex : insee + ssp - MLOps case study to illustrate - open-source project - one-stop-shop - blueprint for building other similar data science platforms

2 Context

2.1 Freins à l'innovation

- Thème général : donner de l'autonomie
- Limites du poste de travail : littérature sur scaling horizontal / vertical
- Observation commune aux différents INS :
  - Insee / SSM : homogénéité des parcours, pourtant grande diversité d'infra, de moyens DSI → difficulté à partager des environnements, des formations → idée de fournir une "sandbox", un commun technologique (2020) [NB : dans la continuité, sandbox à l'échelle européenne via le one-stop-shop (2024)]
  - Visions/incitations différentes DSI/statisticien → sécurité avant le fonctionnel
- Inspirations : DevOps, DataOps

2.2 Innovation technologique

Bearing in mind these limitations, our objective was to develop a data platform empowering statisticians with greater freedom for innovation. To achieve this, we delved into the evolving data ecosystem, identifying two significant trends with the potential to overcome the aforementioned limitations. The first trend signals a move away from traditional big data architectures towards more modular, decoupled structures. The second trend highlights containerization technology as a means to enhance the autonomy of statisticians.

Over the last decade, the landscape of big data has dramatically transformed. Following the publication of Google's seminal papers that introduced the MapReduce paradigm [12, 5], Hadoop-based systems rapidly became the reference architecture of the big data ecosystem, celebrated for their capability to manage extensive datasets through the use of distributed computing. The inception of Hadoop marked a revolutionary step, enabling organizations to process and analyze data at an unprecedented scale. Basically, Hadoop provided companies with all-rounded capabilities for big data analytics : tools for ingestion, data storage (HDFS), and computing capacities (Spark, among others) [8], thus explaining its rapid adoption across industries.

In the late 2010's, Hadoop-based architectures have experienced a clear decline in popularity as the industry shifted toward more flexible, decoupled architectures. In traditional Hadoop environments, storage and compute were co-localized by design : if the source file is distributed across multiple servers (horizontal scaling), each section of the source file is directly processed on the machine hosting that section, so as to avoid network transitions between servers. In this paradigm, scaling the architecture often meant a linear increase in both compute and storage, regardless of the actual demand. In a recent article provocatively titled "Big Data is Dead"[1],

[1] https://motherduck.com/blog/big-data-is-dead/

Jordan Tigani, one of the founding engineers behind Google BigQuery, explains why this model doesn't fit the reality of most data-centric organizations. First, because in practice, "data sizes increase much faster than compute sizes". While the amount of data generated and thus needing to be stored may grow linearly over time, it is generally the case that we only need to query the most recent portions of it, or only some columns and/or groups of rows. Besides, Tigani points out that "the big data frontier keeps receding" : advancements in server computing capabilities and declining hardware costs mean that the number of workloads that don't fit on a single machine - a simple yet effective definition of big data - has been continually decreasing. As a result, by properly separating storage and compute functions, even substantial data processing jobs may end up using "far less compute than anticipated [...] and might not even need to use distributed processing at all".

These insights strongly align with our own observations at Insee in recent years. As a use case of using big data infrastructures to improve statistical processes, a team at Insee set up a Hadoop cluster as an alternative architecture to the one already in use to process sales receipt data in the context of computing the consumer price index. An acceleration of data processing operations by up to a factor of 10 was achieved, for operations that previously took several hours to perform [16]. Despite this great increase in performance, this type of architecture were not reused in subsequent projects for several reasons. Firstly, the architecture proved to be expensive and complex to maintain, necessitating specialized technical expertise rarely found within NSOs [23]. More crucially, we noticed that the needs of recent innovative statistical projects carried out at Insee were very much in line with Tigani's observations. The bottleneck for these projects was generally on the side of computational needs - such as the need for GPUs to train or simply use deep-learning models - rather than storage capacity. Furthermore, although these projects could still deal with substantial data volumes, we observed that effective processing could be achieved using conventional software tools (R, Python) on single-node systems by leveraging recent promising tools from the data ecosystem. First, by using efficient formats to store the data such as Apache Parquet [10], which properties (columnar storage [1], optimisation for the "write once, read many" (WORM) paradigm, ability to partition data, etc.) make it particularly suited to analytical tasks such as those generally performed in official statistics [2]. Second, by performing computations using in-memory computation frameworks such as Apache Arrow [11] or DuckDB [18], that are also based on columnar representation - thus working in synergy with Parquet files - and implementing various optimizations (predicate pushdown, projections pushdown) to limit computations to data effectively needed, enabling much larger-than-memory data processing.

The advent of cloud technologies has been instrumental in facilitating the shift towards decoupled data architectures. Containerization, in particular, encapsulates applications in self-contained environments, ensuring consistency across development, testing, and production. This technology, coupled with orchestrators like Kubernetes, allows for dynamic resource allocation and scaling, reflecting the real-time demands of data processing tasks. Object storage further complements this architec-

ture, offering highly scalable, durable, and cost-effective solutions for data storage that traditional file systems struggle to match.

The rise of containerization also highlights a broader trend toward greater autonomy and agility in software development and deployment, as advocated by the DevOps approach. By abstracting the application from the underlying infrastructure, developers gain the freedom to innovate and iterate rapidly, without being bogged down by environment inconsistencies or deployment complexities.

Observation : convergence d'éco-sytèmes.

Axe : big data is dead → architecture découplage.

- Transition éco big data → éco découplage : co-localisation plus très justifiée
- Stockage objet
- Infra BD tradi très spécialisées (calcul distribué). Aujourd'hui avec ML etc cas d'usages bcp plus diversifiés → outils d'automatisation, MLOPS, GPUs
- Insee : déjà culture fichier SAS + volumétries limitées → sauté l'étape BDD (cf. big data is dead)

Axe : conteneurisation comme moyen d'autonomisation.

- Conteneurisation = light virtualization vs. VM
- Tendance DevOps → DataOps, MLOps
- Reproductibilité des traitements

3 Implementation

3.1 Onyxia

Axe : mise à dispo des technos cloud → favoriser l'autonomie.

- Convergence des choix d'archi. Mais suffisant pour garantir l'autonomie : non → les outils de l'éco-système s'adressent plutôt à des informaticiens (ex : difficulté de configurer Spark sur du stockage objet en mode kube)
- Eco système découplé, mais exigeant → compétences diverses.
- Enjeu : faciliter l'acces aux ressources cloud pour les statisticiens (qui doit déjà s'acculturer à la reproductibilité → convergence avec les outils des développeurs) → double décalage qui demande une assistance
- IHM Onyxia comme liant technique

Axe : principes

- production-ready : outils d'automatisation (-> autonomie)
- no vendor-lockin (enfermement de la structure → coût (licences) et des pratiques → fige les compétences)
- cloud-native : onyxia n'est pas le choix fondamental, le parti pris est sur le choix sous-jacent : conteneurisation + stockage objet

3.2 SSP Cloud

- Orientation plateforme : instance vivante d'Onyxia, ouverte, collaborative, sandbox (cf. ref papier SSP Cloud sur l'aspect plateforme)
- Innovation ouverte → littérature
- Open-data
- Instance de partage : formations reproductibles + utilisation dans les écoles de stats + hackathons (organisation annuelle du funathon cf. one-stop-shop)
- A catalog of services which covers the entire lifecycle of a data science project
- Acculturation aux bonnes pratiques par l'usage

4 A case-study : MLOps APE

This chapter aims, through a concrete example, to illustrate how INSEE managed to deploy its first machine learning model into production. It will delve into the MLOps approach that this project strived to adhere to as much as possible, focusing on the various technologies and infrastructures that were employed. This initial production deployment, while successful, faced various challenges, whether technical or organizational, and we will endeavor to discuss them and propose solutions wherever possible. The idea is to illustrate the development of this project as transparently as possible, without claiming it to be the definitive approach. The entire project is available in open source at https://github.com/orgs/InseeFrLab/teams/codification-ape/repositories and remains under active development.

4.1 Context and motivations

Coding tasks are common operations for all national statistical institutes and can sometimes be challenging due to the size of certain nomenclature. At INSEE, a highly sophisticated coding tool called Sicore was developed in the 1990s to perform various classifications. Sicore uses a reference file that can be considered as a training file, which serves as examples of codings. The label to be coded is compared to the labels contained in the training file, and when the label is recognized, the associated code is assigned. When the label is not recognized, it must be manually classified by an INSEE agent. Two main reasons drove the experimentation of new coding methods. Firstly, there was an internal change with the redesign of the Sirene registry, which lists all companies in France and assigns them a unique identifier, the Siren number, for use by public institutions, notably to improve the daily management of the registry for INSEE agents and to reduce waiting times for companies. Additionally, at the national level, the government launched a one-stop shop for business formalities, allowing more flexibility for business owners in describing their main activities.

The initial testing exercises revealed that Sicore was no longer the suitable tool for performing NACE classification, as only 30% of tasks were being automatically coded. The teams working on the Sirene registry were already overwhelmed with numerous changes, making it unrealistic to further increase their workload with manual reclassification, which is both time-consuming and unstimulating. Therefore, in May 2022, the decision was made to experiment with new methods for performing this classification task, with the aim of using this method in production by January 1, 2023, the launch date of the new Sirene registry, if successful.

This choice of innovation was not initially a voluntary decision but rather a necessity, given that the current state of the process could not remain unchanged. Therefore, all decisions made during this project were taken considering these temporal and organizational constraints. The aim is to present these various strategic choices that we made at INSEE while bearing in mind that they may not be applicable or advisable in all organizations.

Three stakeholders were involved in this project: the business team responsible for managing the Sirene registry, the IT team developing software related to the registry's operation, and the *"innovation"* team tasked with implementing the new coding tool. The latter team is the INSEE Lab, which was created in 2017 with the objective of providing support to other teams on innovation topics to streamline their various projects.

4.2 Démarrage du projet comme les projets expérimental et prise en compte des contraintes

The project we aim to implement is a standard natural language classification problem. Indeed, starting from a textual description, we want to predict the class associated with it in the NACE Rev. 2 nomenclature. This nomenclature has the particularity of being hierarchical and containing 5 different levels[2]: section, division, group, class, and subclass. In total, 732 subclasses exist, which is the level at which we aim to perform our classification. Table 1 summarizes this hierarchical structure with an example.

Level	NACE	Title	Size
Section	H	Transportation and storage	21
Division	52	Warehousing and support activities for transportation	88
Group	522	Support activities for transportation	272
Class	5224	Cargo handling	615
Subclass	**5224A**	**Harbour handling**	**732**

Table 1 NACE Nomenclature

[2] Actually, there are 5 different levels in France but only 4 at the European level.

With the establishment of the one-stop shop, business owners can now freely draft their activity descriptions. As a result, the labels received by INSEE are very different from the harmonized labels that were previously received. Therefore, it was decided to work with machine learning models that have proven their effectiveness in the literature.

4.2.1 1er question, ou on peut travailler ?

- projet ML plusieurs tâches : modularité de l'infra + collaboration (git indispensable, stockage partagé)
- Illustration de la diversité des taches nécessaires dans un projet de ML et modularité indispensable de l'infra utilisée (reprendre infra Big Data trop spécifique et onyxia cool)
- dans notre cas données ouverte donc possibilité d'utiliser le ssp cloud
- Rappeler les contraintes/prérequis que cela impose : utilisation de Git n'est pas aisée et nécessite des formations (mise en place d'un cursus de formateurs pour former à l'Insee), sauvegarde des données sur MinIO et pas en local car environnement éphémère rust bonnes pratiques etc

4.2.2 2eme question, comment travailler ?

- Choix de langage de développement : python. Dire débat R et python, Insee est passé à du tout R mais ecosystème ML plutot python. Ne pas opposer les deux, ils sont complémentaire gnagna
- On travaille sur des notebook en local on obtient des bon résultats mais on arrive rarement à les mettre à l'echelle.
- Rappeler tous les défauts des notebook pour la mise en prod.

4.2.3 3eme question, quel modèle utilisé ?

The model chosen after various trials is the fastText model [14], for several reasons:

1. The innovation team had gained experience in using this model through several previous experiments.
2. The performance obtained was very good.
3. The model is very simple methodologically and quick to train.
4. There is a Java wrapper available that allows reading fastText models. (share github ?)
5. Once trained, the model is lightweight enough to be deployed on our production servers.

rappeler les nouveaux enjeux pour les projets de ML (model versionning, logging parameters) L'utilisation du ssp cloud permet d'accéder à plusieurs logiciels

tous interconnectés pour favoriser le developpement de projet de machine learning favorisant une approche MLOps Objectif d'appliquer cette approche durant ce projet.

4.3 MLflow as the cornerstone of the project

Logiciel qui permet de suivre cette approche = MLflow et c'est dispo sur ssp cloud

- Why Mlflow ?
- Projects
- Models
- Tracking server
- Model registry

4.4 Embracing the power of Onyxia from training to deployment

- Distributing trainings with Argo workflows
- Deployment on the kubernetes cluster (freed from DSI) with fastAPI → conteneurisation Docker
- Automatiser les déploiements avec argoCD

  Environnement dev et production très proche → passage en prod facilité

- Transmission d'une image
- Transmission d'une API

4.5 Monitoring of the model

- Enjeu du monitoring => indispensable
- data drift/ concept drift
- Pour APE : Création d'un dashboard (faire un super graphs qui récap tout)
- encore on utilise les trucs du datalab (argocd pour le déploiement, argoworkflow pour les cronjob quotidien)

4.6 Annotation en continue

- Evaluer la performance en créant un fichier test golden standard -> intégré au dashboard
- Amélioration du jeu d'entrainement en corrigeant les erreurs
- passage en NAF2025 très bientôt gros enjeu

- tout ca réalisé sur le datalab avec LabelStudio
- Rappeler les problèmes rencontrés (faire comprendre aux équipes métiers que c'est ultra important pour améliorer la performance, nécessite ressources humaines importantes..)

4.7 Gouvernance d'un projet de ML/ challenges

5 Discussion

5.1 Future

- Onyxia, un bien commun opensource largement réutilisé (Insee, SSB) → faciliter les contributions pour la postérité du projet open-source, qui dépasse l'Insee
- One-stop-shop : SSP Cloud comme plateforme de référence pour les projets de ML → croissance de l'offre de formation (+ traduction)
- Accompagner les réinstanciations (datafid, POCs dans le secteur privé)
- Multiplication des projets qui passent en prod (applications de dataviz, modèles de ML avec MLOps, webscraping : Jocas/WINs)

5.2 Discussion

- Cout d'entrée important pour l'organisation : stockage objet, cluster kube/conteneurisation
  - Choix fondamental d'archi → limite à la diffusion d'onyxia
  - Assumer le choix : compétences, organisation ...
  - Mais globalement : tendance favorable car beaucoup d'orga et INS font ce choix
- Cout d'entrée important pour le statisticien :
  - Non-persistence de l'environnement → git + stockage objet
  - Travail dans un conteneur → perte de repères sur l'environnement
  - Mais formation : bonnes pratiques + écoles de formation Insee + accompagnements
- SSP Cloud :
  - Instance ouverte → absence de données sensibles → grosse limitation des cas d'usage réalisables + frustrations → en résumé, difficile de maximiser à la fois innovation et sécurité (pb sur-contraint)
  - → résolution via le choix de l'innovation max car sujet des échanges inter-administration de données complexe + le SSP Cloud a pavé la voie à des

instances internes, plus fermées → stratégie assumée "platform-as-a-package" : projet open-source packagé → facilité ++ de réinstanciation
  - Pas une plateforme de diffusion de données → pas de stratégie globale de gouvernance → le sujet de la méta-donnée n'est pas abordé.

- Gouvernance :
  - Quelle organisation ? Equipe DS centralisée qui vient en appui ou data scientists dans les orgas métiers ? Collaboration avec les équipes infos ? (cf. graphique orga/compétences de Romain)

Appendix

References

1. Daniel Abadi, Peter Boncz, Stavros Harizopoulos, Stratos Idreos, Samuel Madden, et al. The design and implementation of modern column-oriented database systems. *Foundations and Trends® in Databases*, 5(3):197–280, 2013.
2. Abdullah I Abdelaziz, Kent A Hanson, Charles E Gaber, and Todd A Lee. Optimizing large real-world data analysis with parquet files in r: A step-by-step tutorial. *Pharmacoepidemiology and Drug Safety*, 2023.
3. Afshin Ashofteh and Jorge M Bravo. Data science training for official statistics: A new scientific paradigm of information and knowledge development in national statistical systems. *Statistical Journal of the IAOS*, 37(3):771–789, 2021.
4. Thomas H Davenport and DJ Patil. Data scientist. *Harvard business review*, 90(5):70–76, 2012.
5. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
6. Pascaline Descy, Vladimir Kvetan, Albrecht Wirthmann, and Fernando Reis. Towards a shared infrastructure for online job advertisement data. *Statistical Journal of the IAOS*, 35(4):669–675, 2019.
7. DGINS. Bucharest memorandum on official statistics in a datafied society. https://ec.europa.eu/eurostat/documents/13019146/13237859/The+Bucharest+Memorandum+on+Trusted+Smart+Statistics+FINAL.pdf/7a8f6a8f-9805-e77c-a409-eb55a2b36bce?t=1634144384767, 2018.
8. Bijesh Dhyani and Anurag Barthwal. Big data analytics using hadoop. *International Journal of Computer Applications*, 108(12):0975–8887, 2014.
9. EUROSTAT. Essnet big data 2 - final technical report. https://wayback.archive-it.org/12090/20221110013641/https://ec.europa.eu/eurostat/cros/system/files/wpa_deliverable_a5_final_technical_report_2021_06_29.pdf, 2021.
10. Apache Software Foundation. Apache parquet. https://parquet.apache.org/, 2013.
11. Apache Software Foundation. Apache arrow. https://arrow.apache.org/, 2016.
12. Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 29–43, 2003.
13. Taeke Gjaltema. High-level group for the modernisation of official statistics (hlg-mos) of the united nations economic commission for europe. *Statistical Journal of the IAOS*, 38(3):917–922, 2022.
14. Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

15. Alexander Kowarik and Magdalena Six. Quality guidelines for the acquisition and usage of big data with additional insights on web data. In *4th International Conference on Advanced Research Methods and Analytics (CARMA 2022)*, pages 269–269. Editorial Universitat Politècnica de València, 2022.
16. Marie Leclair et al. Utiliser les données de caisses pour le calcul de l'indice des prix à la consommation. *Courrier des statistiques*, 3:61–75, 2019.
17. Ling Liu. Computing infrastructure for big data processing. *Frontiers of Computer Science*, 7:165–170, 2013.
18. Mark Raasveldt and Hannes Mühleisen. Duckdb: an embeddable analytical database. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1981–1984, 2019.
19. Fabio Ricciato, Freddy De Meersman, Albrecht Wirthmann, Gerdy Seynaeve, and Michail Skaliotis. Processing of mobile network operator data for official statistics: the case for public-private partnerships. In *104th DGINS conference*, 2018.
20. Fabio Ricciato, Albrecht Wirthmann, Konstantinos Giannakouris, Michail Skaliotis, et al. Trusted smart statistics: Motivations and principles. *Statistical Journal of the IAOS*, 35(4):589–603, 2019.
21. Anam Saiyeda and Mansoor Ahmad Mir. Cloud computing for deep learning analytics: A survey of current trends and challenges. *International Journal of Advanced Research in Computer Science*, 8(2), 2017.
22. David Salgado, Luis Sanguiao-Sande, Sandra Barragán, Bogdan Oancea, and milena suarez castillo. A proposed production framework with mobile network data. 11 2020.
23. Steven Vale. International collaboration to understand the relevance of big data for official statistics. *Statistical Journal of the IAOS*, 31(2):159–163, 2015.