NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

BSc THESIS

A data analytics framework for COVID-19 analysis

Michail V. Tatas

Supervisor: **Alexis Delis,** Professor

ATHENS

OCTOBER 2022

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ένα πλαίσιο ανάλυσης δεδομένων για την ανάλυση του Κορονοϊού

Μιχαήλ Β. Τατάς

Επιβλέπων: Αλέξης Δελής, Καθηγητής

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2022

BSc THESIS

A data analytics framework for COVID-19 analysis

Michail V. Tatas

S.N.: 1115201700161

SUPERVISOR: **Alexis Delis,** Professor

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ένα πλαίσιο ανάλυσης δεδομένων για την ανάλυση του Κορονοϊού

Μιχαήλ Β. Τατάς
Α.Μ.: 1115201700161

ΕΠΙΒΛΕΠΩΝ: **Αλέξης Δελής**, Καθηγητής

ABSTRACT

Nowadays, the tremendous increase of data has contributed to the rise of a "data-driven" era, in which big data analytics are used in every sector (agriculture, health, energy, infrastructure, economics and insurance, sports, food, transportation) and every world economy. The problem we are currently facing is that traditional data analyses techniques and systems most of the times are not able to handle such large quantities of data and even if they can, they take a lot of time doing so, which in a such fast paced world is simply not acceptable, let alone mentioning real time systems where data must be consumed and analysed in a matter of minutes at most.

The critical question that arises from these developments is, how to develop a highly performant system to efficiently analyze big data and mine them in order to extract useful information. In order to dive deep in this issue in this Bachelor Thesis we provide the implementation of a scalable and efficient system, with the use of tools such as Kubernetes, Linux Containers and Spark, able to handle big data and we put the system to the test using it for analysing the COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University, the World Healthcare Organization (WHO) vaccinations data set and the European Statistical Organization (Eurostat) education data set and extracting some pretty interesting insights.

SUBJECT AREA: Data Engineering and Data Science

KEYWORDS: Data Engineering, Data Science, COVID-19, Spark, Kubernetes, Linux Containers

ΠΕΡΙΛΗΨΗ

Στις μέρες μας, η τεράστια αύξηση των δεδομένων έχει συμβάλει στην άνοδο μιας "data- driven" εποχής, στην οποία η ανάλυση μεγάλων δεδομένων χρησιμοποιείται σε κάθε τομέα (γεωργία, υγεία, ενέργεια, υποδομές, οικονομία και ασφάλιση, αθλητισμός, τρόφιμα, με- ταφορές) και σε κάθε παγκόσμια οικονομία. Το πρόβλημα που αντιμετωπίζουμε σήμερα είναι ότι οι παραδοσιακές τεχνικές και τα συστήματα ανάλυσης δεδομένων τις περισσότε- ρες φορές δεν είναι σε θέση να διαχειριστούν τόσο μεγάλες ποσότητες δεδομένων, που ακόμη και αν μπορούν, χρειάζονται πολύ χρόνο για να το κάνουν, κάτι που σε έναν κόσμο που κινείται τόσο γρήγορα είναι απλά μη αποδεκτό, πόσο μάλλον όταν αναφερόμαστεσε συστήματα πραγματικού χρόνου, όπου τα δεδομένα πρέπει να καταναλώνονται και να αναλύονται το πολύ σε λίγα λεπτά.

Το κρίσιμο ερώτημα που προκύπτει από αυτές τις εξελίξεις είναι, πώς μπορεί να αναπτυ- χθεί ένα σύστημα υψηλής απόδοσης για την αποτελεσματική ανάλυση μεγάλων δεδομέ- νων και την εξόρυξή τους προκειμένου να εξαχθούν χρήσιμες πληροφορίες. Προκειμένου να εμβαθύνουμε σε αυτό το θέμα σε αυτή τη Διπλωματική Εργασία παρέχουμε την υλο- ποίηση ενός κλιμακούμενου και αποδοτικού συστήματος, με τη χρήση εργαλείων όπως το Kubernetes, τα Linux Containers και το Spark, ικανού να διαχειριστεί μεγάλα δεδομένα και εξετάζουμε το σύστημα χρησιμοποιώντας το για την ανάλυση του συνόλου δεδομένων COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University, του συνόλου δεδομένων εμβολιασμών κατα του Κορονοϊού του Παγκόσμιου Οργανισμού Υγείας (WHO) και του συνόλου δεδομένων για την εκπαί- δευση, του Ευρωπαϊκού Στατιστικού Οργανισμού (Eurostat) και τελικα εξάγουμε μερικές αρκετά ενδιαφέρουσες πληροφορίες.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μηχανική και Επιστήμη Δεδομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μηχανική Δεδομένων, Επιστήμη Δεδομένων, Κορονοϊός, Spark,Kubernetes, Linux Containers

ACKNOWLEDGEMENTS

I would like to thank my supervisor Mr. Alexis Delis for his excellent guidance and assistance for the preparation of this undergraduate thesis

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Data is characterized as the lifeblood of decision making and the raw material for accountability. Without high quality data providing the right information on the right things at the right time, designing, monitoring and evaluating effective policies becomes almost impossible. In that context, an ongoing attention to data and data-driven approaches from academics and professionals exist, since the knowledge arising from data analysis processes leads to the promotion of innovative activity, transforming organizations, enterprises and national economies.

In this data-driven era, many enterprises independent of size, from start-ups to large organizations, attempt to obtain data-driven culture struggling for competitive advantage against rivals. Nowadays managerial decisions rely on data-based analytics and less on the leaders experience and enterprises aim to leverage data generated within organizations through their operations to gain valuable insights for better, faster and more accurate decisions in crucial business issues and .

The importance of data is very clear by now, but in order to gain insights and valuable information, raw data need to be analysed to extract information and make conclusions. This is the reason we need systems able to undertake this task, but due to the sheer scale of data nowadays these systems need to be highly performant, so they can analyse data efficiently and in a timely manner. Having this in mind we wanted to implement a very scalable system combining Kubernetes, which is not yet very popular in data engineering and data science, with one of the most popular data engineering tools named Spark. This combination brings together the best of both worlds, because Spark is great at distributing work over a cluster of machines, but is not able to manage them, something in which Kubernetes is great at.

In order to test our system and whether it can handle big loads of data we used the COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University, the World Healthcare Organization (WHO) vaccinations data set and the European Statistical Organization (Eurostat) education data set. Finally in chapter 5 we showcase some very interesting insights, regarding how a country's vaccination rate is correlated to its education rate. After analysing the data and using the Spearman and Pearson correlation coefficients, we came to the conclusion that these two datasets have a close correlation and when a country's participation in education is lower percentage, then it is expected that its participation in the vaccinations follows the same pattern.

2. PREREQUISITE KNOWLEDGE

There are a few tools that the reader has to be familiar with before we move on to the real system and its implementation. If there is anything you are already familiar with, feel free to skip it.

2.1 Linux Containers

A Linux Container [1] is a set of one or more processes that are isolated from the rest of the system. All the files necessary to run them are provided from a distinct image, meaning Linux Containers are portable and consistent as they move from development, to testing and finally to production. This makes them much quicker to use than development pipelines that rely on replicating traditional testing environments.
In a way, containers behave like a virtual machine. To the outside world, they can look like their own complete system. But unlike a virtual machine, rather that creating a whole virtual operating system, containers don't need to replicate an entire operating system, only the individual components they need in order to operate. This gives a significant performance boost and reduces the size of the application. They also operate much faster, as unlike traditional virtualization the process is essentially running natively on its host, just with an additional layer of protection around it.

2.2 Kubernetes

2.2.1 What is Kubernetes

Kubernetes (K8s) [2] is a portable, extensible, open source platform for managing containerized workload and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem.
K8s is an abbreviation that results from counting the eight letters between the "K" and the "s".

2.2.2 Why is Kubernetes useful

In order to understand why Kubernetes is useful we need to go back in time.

Traditional deployment era : Early on, organizations ran applications on physical servers. There was no way to define resource boundaries for applications in a physical server, and this caused resource allocation issues. For example, if multiple applications run on a physical server, there can be instances where one application would take up most of the resources, and as a result, the other applications would underperform. A solution for this would be to run each application on a different physical server. But this did not scale as resources were underutilized, and it was expensive for organizations to maintain many physical servers.

Virtualized deployment era : As a solution, virtualization was introduced. It allows you to run multiple Virtual Machines (VMs) on a single physical server's CPU. Virtualization allows applications to be isolated between VMs and provides a level of security as the information of one application cannot be freely accessed by another application.

Virtualization allows better utilization of resources in a physical server and allows better scalability because an application can be added or updated easily, reduces hardware costs, and much more. With virtualization you can present a set of physical resources as a cluster of disposable virtual machines.

Each VM is a full machine running all the components, including its own operating system, on top of the virtualized hardware.

Container deployment era : Containers are similar to VMs, but they have relaxed isolation properties to share the Operating System (OS) among the applications. Therefore, containers are considered lightweight. Similar to a VM, a container has its own filesystem, share of CPU, memory, process space, and more. As they are decoupled from the underlying infrastructure, they are portable across clouds and OS distributions.

Containers have become popular because they provide extra benefits, such as:

- Agile application creation and deployment: increased ease and efficiency of container image creation compared to VM image use.
- Continuous development, integration, and deployment: provides for reliable and frequent container image build and deployment with quick and efficient rollbacks (due to image immutability).
- Dev and Ops separation of concerns: create application container images at build/release time rather than deployment time, thereby decoupling applications from infrastructure.
- Observability: not only surfaces OS-level information and metrics, but also application health and other signals.
- Environmental consistency across development, testing, and production: Runs the same on a laptop as it does in the cloud.
- Cloud and OS distribution portability: Runs on Ubuntu, RHEL, CoreOS, on-premises, on major public clouds, and anywhere else.
- Application-centric management: Raises the level of abstraction from running an OS on virtual hardware to running an application on an OS using logical resources.
- Loosely coupled, distributed, elastic, liberated micro-services: applications are broken into smaller, independent pieces and can be deployed and managed dynamically – not a monolithic stack running on one big single-purpose machine.
- Resource isolation: predictable application performance.
- Resource utilization: high efficiency and density.

2.3 Apache SPARK

Apache Spark [3] is an open source data processing framework than can quickly perform processing tasks on very large data sets, and also can distribute data processing tasks

across multiple clustered computers, either on its own or in tandem with other distributed computing tools. These two qualities are key to the world of big data and machine learning, which require the marshalling of massive computing power to crunch through large data stores. Spark also takes some of the programming burdens of these tasks off the shoulders of developers with an easy to use API(Java, Scala, Python and R) that abstracts away much of the grunt work of distributed computing and big data processing.
From its humble beginnings in the AMPLab at U.C. Berkeley in 2009, Apache Spark has become one of the key big data distributed processing frameworks in the world. Spark can be deployed in a variety of ways, provides native bindings for the Java, Scala, Python and R programming languages, and supports interactive queries with SQL, streaming data, batch data, machine learning and graph processing.

2.4 PostgreSQL

PostgreSQL [4] is an advanced, enterprise class open source relational database that supports both SQL (relational) and JSON (non-relational) querying. It is a highly stable database management system, backed by more than 20 years of community development which has contributed to its high levels of resilience, integrity, and correctness. PostgreSQL is used as the primary data store or data warehouse for many web, mobile, geospatial, and analytics applications. The latest major version is PostgreSQL 15.

PostgreSQL has a rich history for support of advanced data types, and supports a level of performance optimization that is common across its commercial database counterparts, like Oracle and SQL Server.

2.5 Apache Superset

Apache Superset [5] is an open-source software application for data exploration and data visualization able to handle data at petabyte scale. The application started as a hackathon project by Maxime Beauchemin (creator of Apache Airflow) while working at Airbnb and entered the Apache Incubator program in 2017. In addition to Airbnb, the project has seen significant contributions from other leading technology companies, including Lyft and Dropbox. Superset graduated from the incubator program and became a top-level project at the Apache Software Foundation in 2021.

3. SYSTEM DESIGN

Apache Spark is an open-source distributed computing framework, with which data scientists and engineers may implement programs that process enormous volumes of data in a few lines of code with Spark handling parallelizing the job across a cluster of machines.
On the other hand Apache Spark doesn't itself handle those machines. It requires a cluster manager (scheduler) with the options been shown below :

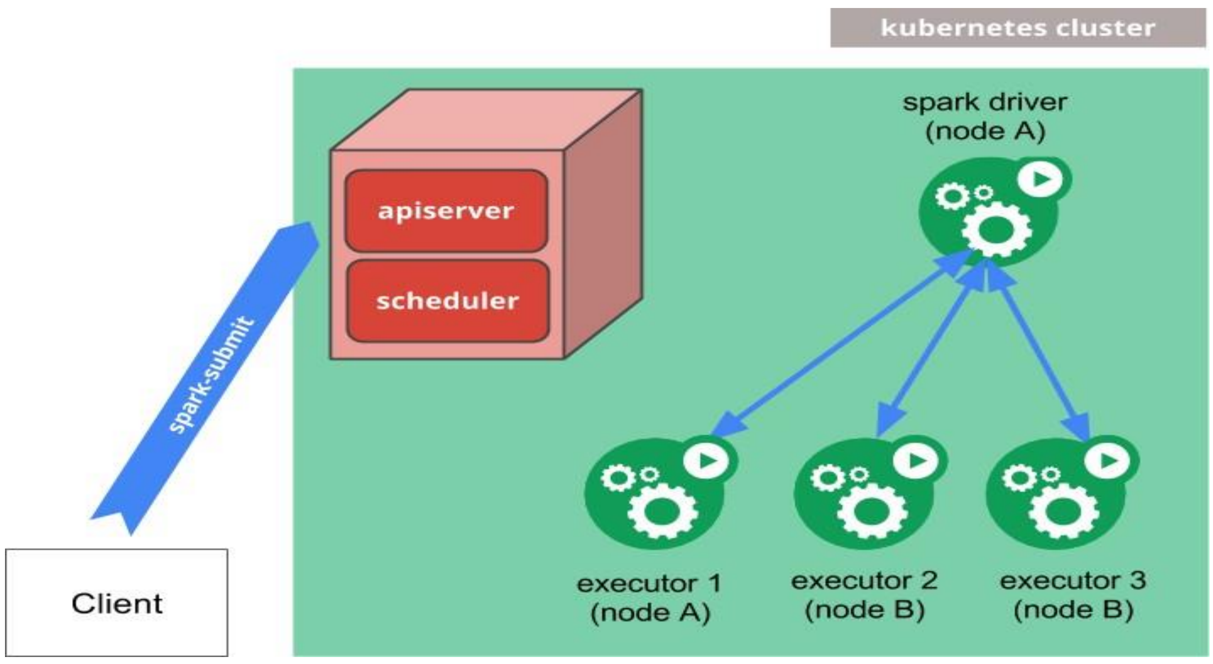
- Standalone : Simple cluster manager, with limited features and it comes prepacked with Spark.

- Apache Mesos : An open-source cluster manager, which was very popular for big data workloads, but is now in decline over the last years

- Hadoop YARN : The JVM-based cluster manaer of hadoop, which is the most wide-spread cluster manager for Spark, both for deployments on premises and on cloud.

- Kubernetes : Apache Spark runs natively on Kubernetes since its 2.3 version. This option is gaining massive traction very quickly. As of 2021 Spark on Kubernetes has been officialy declared as generally available and production ready.

3.1 How Apache Spark on Kubernetes works



Apache Spark can be used to submit a Spark application directly to a Kubernetes cluster. the submission mechanism works as follows :

- Spark creates a Spark driver running within a Kubernetes pod.

- The driver creates executors which are also running within Kubernetes pods and connects to them, and executes applications code. If dynamic allocation is enabled the number of Spark executors dynamically evolves based on load, otherwise it is a static number.

- When the application completes, the executor pods terminate and are cleaned up, but the driver pod persists logs and remains in "completed state" in the Kubernetes API until it is eventually garbage collected or manually cleaned up.

3.2 Advantages of using Apache Spark with Kubernetes

Running Apache Spark on Kubernetes has a lot of important advantages that are listed below:

- Containerization : This is the main motivation for using Kubernetes itself and all the huge hype surrounding it. Containerization brings a lot of advantages not only to traditional software engineering but on big data as well. Containers make the application more portable (build your dependencies once, run everywhere), they simplify dependencies and they enable repeatable and reliable workflows. This is especially important for Spark since it's dependency management is notoriously painful. This way you can use a different docker image for each application, with all the dependencies you need.

- Efficient resource sharing and isolation : While using YARN, if you want to reuse the same cluster for concurrent Sparks applications (for less cost), you will have to make a compromise on the isolation. All the applications will share the same Spark and Python version, share the same libraries and environments. On top of that if one big job is being run, other jobs are very likely to run slower.
  Using Kubernetes as a cluster manager will set you free of those issues, applications will be fully isolated, so you can choose the versions of the libraries you want and K8s will provide you with powerful resource sharing.

- Cloud Agnostic : Deploying Spark on K8s makes the application very agnostic and it is very easy to migrate from cloud vendor to cloud vendor and even on premises.

- Apache Spark with Kubernetes performance : Apache Spark on K8s has caught up with YARN and there is no performance difference

3.3 Disadvantages of using Apache Spark with Kubernetes

- Spark on Kubernetes at scale requires time and expertise : If someone is new to Kubernetes it can be difficult and even frightening to handle, with all the new complexity it introduces and take you away from you real goal. Of course even if you have K8s experience there are a lot of things to be configured like creating and configuring the Kubernetes cluster and the node pools, setting up the Spark operator and the autoscaler, setting up the logs, monitoring and security etc.

- Spark should be run with its latest versions : In order to benefit from Spark on K8s you have to run Spark with greater version than 2.3 (version in which K8s was officially supported by Spark) and even better with Spark version greater than 3.0 since earlier versions although they support K8s, they lack some critical features.

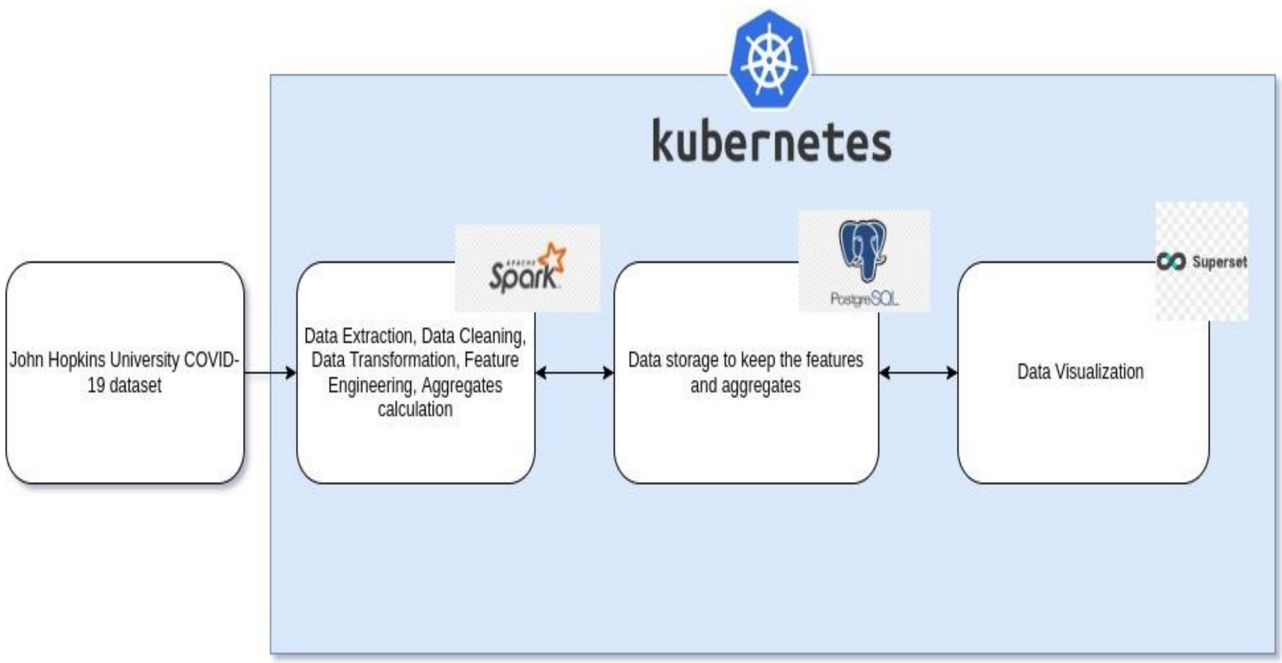
3.4 System Implementation

For our system implementation we have chosen to use Spark on Kubernetes, since it doesn't lack in performance compared to the other cluster managers that work with Spark. Another big argument supporting this decision is that this system scales extremely well with the help of Kubernetes and containers. So all the data analytics we perform on the datasets are being implemented in PySpark, which is an interface for Apache Spark in Python and allows the programmer to write applications using Python APIs, but also provides the PySpark shell for interactively analyzing the data in a distributed environment. This means that we use the python APIs and SQL to analyse the data and Spark calculates everything in distributed fashion, splitting the work in parts and assigning them to a number of executors, which number is decided after communication with Kubernetes. From there on the actual executors are implemented and managed by Kubernetes, using its pods.

This provides us with very fast and efficient computing since, depending on the settings we provide, Kubernetes can scale out as much as we want and scale back in when computing power is not needed, while always communicating with Spark to decide how many executors are needed during any given moment and respecting the settings we have provided. The settings we are referring to are that, we can provide a maximum and minimum number of executors alive at all times, or we can completely let K8s and Spark free and they can decide how many executors are needed and this tactic can scale infinitely if we do not provide an upper limit, especially in a cloud environment.

To store the results that occur from the data analysis we use a PostgreSQL, an open source database, which has been setup in its own separate pod and is set to have 3 containers at all times for high availability. Lastly in a separate pod Apache Airflow is set up, which is an open source data visualization tool, although for data visualization purposes we also used the library Folium, which makes it easy to visualize data that have been manipulated in Python.

The architecture of the system is being shown below :


kubernetes
Spark
PostgreSQL
Superset
John Hopkins University COVID-19 dataset
Data Extraction, Data Cleaning, Data Transformation, Feature Engineering, Aggregates calculation
Data storage to keep the features and aggregates
Data Visualization

4. DATA SETS AND DATA CLEANING

The first data-set used is the **COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University**] [6]. It contains the following columns :

- FIPS : US only. Federal Information Processing Standards code that uniquely identifies counties within the USA.
- Admin2 : County name. US only.
- Province_State : Province, state or dependency name.
- Country_Region : Country, region or sovereignty name. The names of locations included on the Website correspond with the official designations used by the U.S. Department of State.
- Last_Update : MM/DD/YYYY HH:mm:ss (24 hour format, in UTC).
- Lat and Long_ : Dot locations on the dashboard. All points (except for Australia) shown on the map are based on geographic centroids, and are not representative of a specific address, building or any location at a spatial scale finer than a province/state. Australian dots are located at the centroid of the largest city in each state.
- Confirmed : Counts include confirmed and probable (where reported).
- Deaths : Counts include confirmed and probable (where reported).
- Incident_Rate : Incidence Rate = cases per 100,000 persons.
- Case_Fatality_Ratio (%) : Case-Fatality Ratio (%) = Number recorded deaths / Number cases.

The second data-set used is the **World Healthcare Organization (WHO) vaccinations data set**
citewhoData. It contains the following columns :

- Country
- Iso3
- WHO_Region
- Data_Source
- Date_Updated
- Total_Vaccinations
- Persons_Fully_Vaccinated

The third data-set used is the **European Statistical Organization (Eurostat) education data set** [7]. It contains the following columns :

- Country
- Population
- Primary Education
- Upper Second Education
- Tertiary Education

DATA CLEANING

The different datasets had to be cleaned in order to be used in experiments so the following were done:

- **Country and Row dropping** : While reviewing some statistics that we produced we noticed that North Korea's death to confirmed percentage was 6200%, with 1 confirmed case and 62 confirmed deaths, and MS Zaandam's was 22.2%, with 9 confirmed and 2 deaths. Of course these results occured due to these countries not properly reporting their confirmed cases and the confirmed deaths from COVID-19 and had to be removed completely from the dataset. Similarly we dropped some columns that contained the value 0, nan and null.
- **Column name misspelling** : A lot of files on the Covid-19 cases folder had misspelled the name of the columns so in order to face this issue, a python script had to be created that checked every file and found every file that had wrong column names and then run once more and changed every column name to the correct one and finally concatenated all the files together so they will be easier to handle during the data analysis process.
- **Country name Standardization** : Every dataset had different names for the same countries, for example one would have the name "The United Kingdom" and another would have the name "United Kingdom", or one dataset would have the name "Czechia" and the other "Czech Republic". In order to face this issue we had to standardise the country naming between the datasets based on the expected country names from the visualization librabry folium and from the Apache Superset and so we changed every country name to the corresponding one in the new standard. Furthermore all of the datasets had entries, where the name of the country would be misspelled like "UnitedKingdom", where it is missing a space, or "NorthMacedonia" with the same problem and a variety of other mistakes and misspellings that needed to be correcteed.
- **Extra symbols removal** : In the education dataset from EUROSTAT, a lot of values were accompanied with symbols such as "(z)", "(u)", etc., which needed to be removed. This took a bit of effort because it was not standard and not all the values had it next to them. Also many values happened to have two symbols in a row "% %", which was also an issue. As a result, this process was hard to automate, so it was done semi-automated, by observing every symbols that needed to be removed and removing it with a function, but this was a long process, because some of the so called symbols were actually letters which led to this process being more difficult.

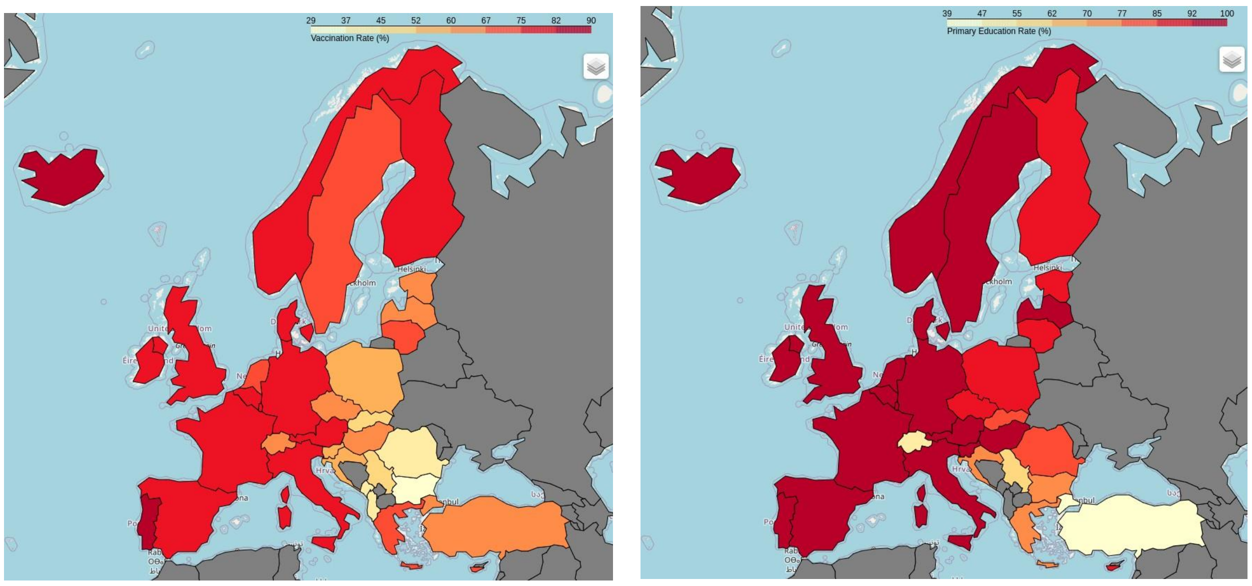
- **Numbers restructuring** : In the education data-set numbers were presented with spaces between the hundreds, thousands, millions (for example 1 215 290.0) and this had to be changed so that it would not contain all the spaces, because the numbers in this form were not recognised by the SQL queries.

5. EXPERIMENTS

One of the most interesting experiments done during our analysing of the data was the correlation between a country's vaccination rate and it's educational level. The hypothesis that the vaccination rate of a country would be correlated to its education rate was made and we present the data of this comparison by plotting the data for the vaccination rate and the data for the education rate for the whole of Europe and after that we use to functions two calculate if there is any correlation between the two datasets. The education is split in three parts : Primary, Secondary and Tertiary.

Firstly we are plotting the primary education data and the vaccination rate and we notice that, as we can see from the images below a lot of the darker shades in the left image that contains the vaccination rate correspond to darker shades in the right image as well that contains the primary education rate, giving us the clue that our hypothesis is most probably true.

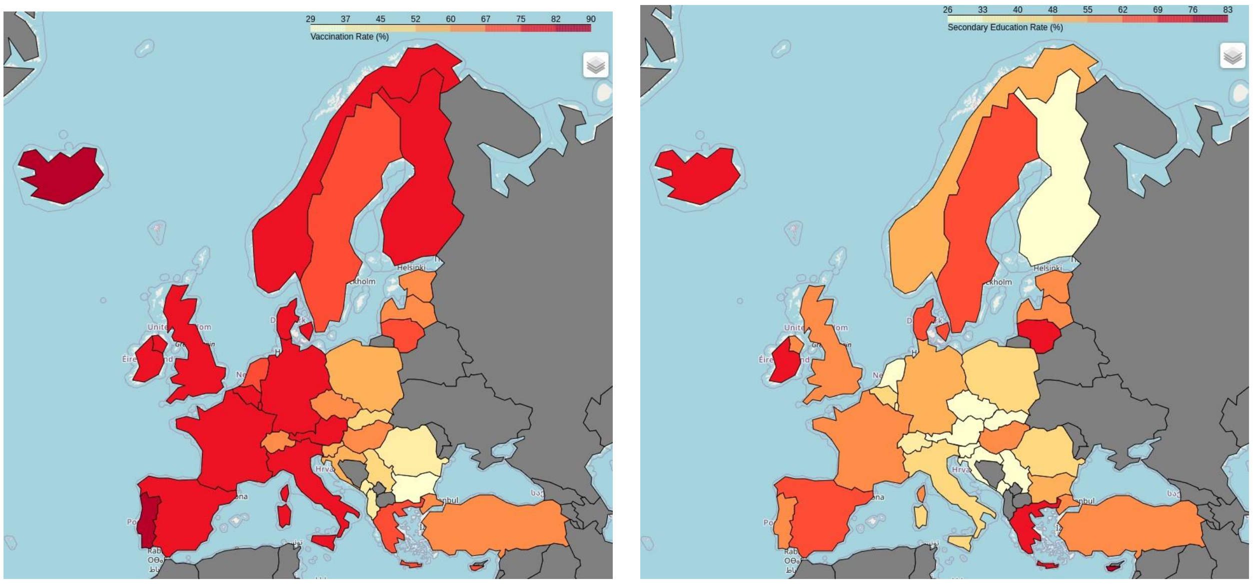
5.1 Data Visualizations



(a) Vaccination Rate (b) Primary Education Rate

Figure 5.1: Vaccination Rate and Primary Education Rate Comparison

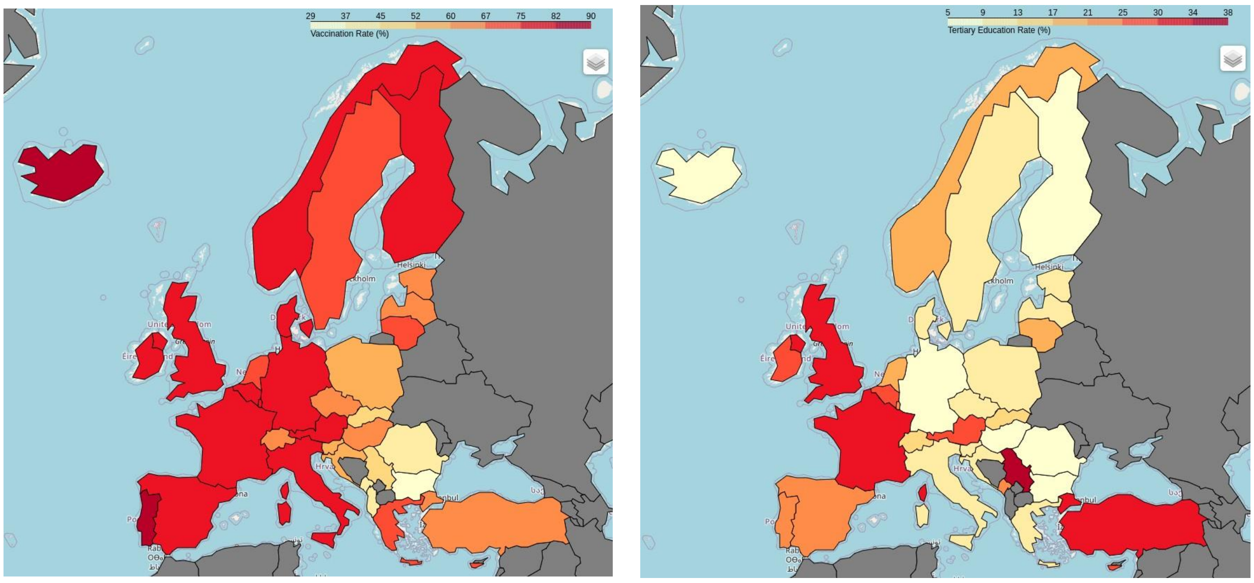
Secondly we are plotting the secondary education data with the vaccination rate and the same as above can be said for the images below although the correlation between left image(vaccination rate) with the right image(secondary education) seems weaker than that depicted in the previous images, but still we notice that a lot of the darker shades on the vaccination rate correspond to darker shades in the secondary education rate.



(a) Vaccination Rate (b) Secondary Education Rate

Figure 5.2: Vaccination Rate and Secondary Education Rate Comparison

Last but not least we plot the tertiary education level data with the vaccination rate and the correlation of the shades in the two images is hard to notice with the naked eye, if there is any correlation at all.



(a) Vaccination Rate (b) Tertiary Education Rate

Figure 5.3: Vaccination Rate and Tertiary Education Rate Comparison

5.2 Data Correlation Methods

Of course we can not assume correlation between the vaccination rate and the different levels of education just by noticing the plotted data, that's why we used the Spearman's and the Pearson's correlation method's in order to mathematically prove their correlation if any.

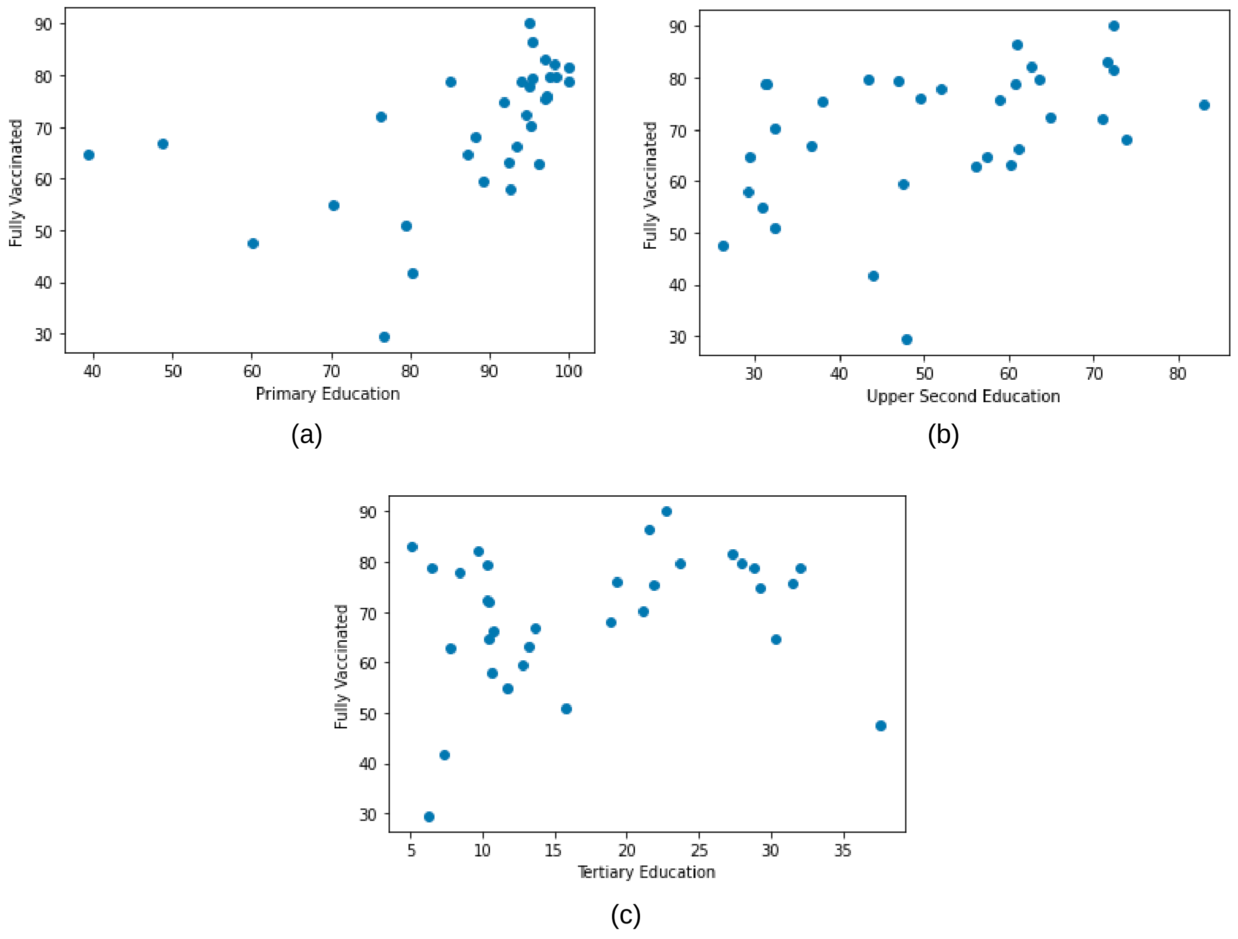


Figure 5.4: Scatter plot between different education level data and vaccination rate

Tha Spearman rank correlation coefficient or Spearman's p, named after Charles Spearman, is a nonparametric measure of rank correlation (statistical dependence between the rankings of two variables). It assesses how well the relationship between two variables can be described using a monotonic function.

The Pearson correlation coefficient, also knows as Pearson's r, the bivariate correlation is a measure of linear correlation between two sets of data. It is defined as the ratio between the covariance of two variables and the produck of their standard deviations. Thus, it is essentially a normalized measurement of the covariance, such that the result always has a value between -1 and 1. As with covariance itself, the measure can only reflect a linear correlation of variables and ignores many other types of relationships or correlations.

It is visible from the scatter plots above, produced by plotting each education level's data with the vaccination rate, that the data distribution is not Gaussian. This means that the Spearman's correlation method is the better one to choose in this case since it is meant to be used for non-Gaussian data. Despite that we will use the Pearson's correlation method as well, since it is not irrelevant,but the main focus should be given to the Spearman's correlation method and all the conclusions we draw take it more into consideration.

5.2.1 Primary education - Vaccination Rate

The Spearman correlation coefficient outputs the result of 0.72, while the Pearson correlation coefficient output the result of 0.51. This clearly means that there is significant correlation between these two datasets and implies that when a country has lower participation in primary education then it tends to have a lower vaccination rate.

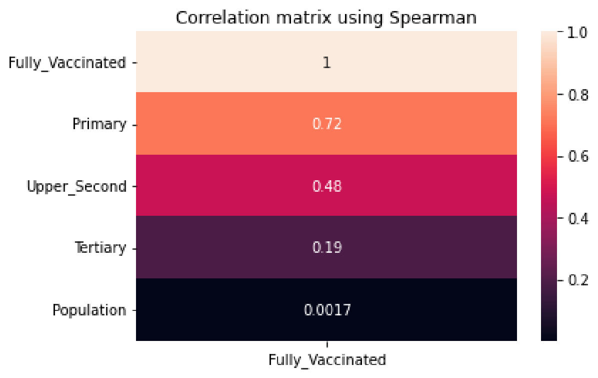
5.2.2 Secondary education - Vaccination Rate

The Spearman correlation coefficient outputs the result of 0.48, while the Pearson correlation coefficient output the result of 0.43. These results still suggest that there is correlation to some extend between these two datasets and that participation in secondary education affect the vaccination rate although not as strongly as the primary education level.

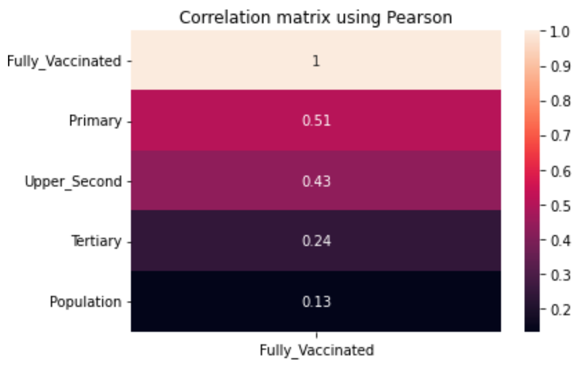
5.2.3 Tertiary education - Vaccination Rate

The Spearman correlation coefficient outputs the result of 0.19, while the Pearson correlation coefficient output the result of 0.24. This suggests that there is almost no correlation between these two datasets.

In conclusion we can safely say that vaccination rate does depend on the education level of the country, especially at the primary education level. Furthermore secondary education level does affect the vaccination rate as well, although not to the extend the primary education does. In addition to that it seems that only the most basic education levels have a big impact on the vaccination percentage of a country, while tertiary education does not seem to have almost any correlation to it.



(a)



(b)

Figure 5.5: Spearman's and Pearson's Correlation Coeffients

6. CONCLUSIONS

In conclusion we created a highly scalable data analysis system containerizing Spark applications in Linux Containers and managing them with Kubernetes. This system can perform data analytics very efficiently and fast, since Spark distributes the workload of computing the data analysis tasks and assigns part of the computation to as many executors as needed, while the user just has to define some settings and then the whole communication part between the tools is handled by the tools themselves. Furthermore after trying different analytics and tools we think that this framework has huge potential, as except its scalability it solves another notorious problem of data engineering and data science, which is portability, since the different tools in these fields depend on a lot of common libraries and many times conflicts appear with different versions of libraries that are needed by more than one tool. Of course this issue does not exist in our system, since if developed properly every tool can be containerized and be separate from the others, being able to use whatever dependency it has without the fear of a conflict.

Regarding the analytics we performed, using this system we analysed the COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University, the World Healthcare Organization (WHO) vaccinations data set and the European Statistical Organization (Eurostat) education data set. After digging into the data, we came up with the hypothesis that the education level of a country, could somehow affect its vaccination rate. Keeping this in mind we created the appropriate datasets and taking their analysis into account and using different statistic tools such as the Spearman and Pearson correlation coefficients to back up mathematically our results, we come to the conclusion that the percentage of vaccinated people is directly affected by the percentage of the population that participated mainly in primary education and to some extend in secondary education as well.

ABBREVIATIONS - ACRONYMS

K8s	Kubernetes
API	Application Programming Interface
WHO	World Health Organization
EUROSTAT	European Statistics Agency
CSSE	Center for Systems Science and Engineering
SQL	Structured Query Language

BIBLIOGRAPHY

[1] Redhat. “What’s a Linux container?” In: Updated May 11, 2022. URL: https://www.redhat.com/en/topics/containers/whats-a-linux-container.

[2] Kubernetes. “Kubernetes Overview”. In: Last modified July 25, 2022. URL: https://kubernetes.io/docs/concepts/overview/.

[3] Apache Spark. “Arache Spark Overview”. In: URL: https://spark.apache.org/ docs/latest/.

[4] PostgreSQL. “What Is PostgreSQL”. In: URL: https://www.postgresql.org/docs/ 15/intro-whatis.html.

[5] Apache Superset. “Apache Superset Introduction”. In: URL: https : / / superset . apache.org/docs/intro/.

[6] Ensheng Dong, Hongru Du and Lauren Gardner. “An interactive web-based dash- board to track COVID-19 in real time”. In: vol. 20. 5. 2020, pp. 533–534. DOI: https://doi.org/10.1016/S1473-3099(20)30120-1. URL: https://www.sciencedirect.com/science/article/pii/S1473309920301201.

[7] Eurostat. “Population by educational attainment level, sex and NUTS 2 regions”.In: 2019. URL: https://ec.europa.eu/eurostat/cache/RCI/#?vis=nuts1.education&lang=en.