

# Fuel Price Streaming Data in France

**Flihi Arij**

Higher School of Communication of Tunis  
arij.flihi@supcom.tn

**Sadkaoui Marwa**

Higher School of Communication of Tunis  
marwa.sadkaoui@supcom.tn

## **Abstract**

*Streaming data, also known as real-time data, is data that is generated continuously and made available for analysis as it is collected. It can be used for a wide range of applications, real-time analytics: streaming data can be analyzed in real-time to identify patterns and trends, making it useful for applications such as fraud detection, recommendation systems, and anomaly detection. This study shows streaming data pipeline of fuel price in the region of France.*

*Fuel price streaming data in France refers to real-time or near real-time information on the prices of various types of fuel, such as gasoline and diesel, at different fuel stations throughout the country. This data can be used by consumers to make informed decisions about where to fill up their vehicles, by fuel retailers to monitor and adjust their prices, and by government bodies and researchers to study trends in fuel pricing and consumption. It can be collected through a variety of means, including scraping the websites of fuel retailers or using sensor networks installed at fuel stations. The data can then be analyzed using a variety of methods, including machine learning, to extract insights and make predictions about future fuel prices. This github repository contains the finished code: <https://github.com/arige160/StreamingFuelPriceData>*

**Keywords:** Stream data; Fuel price; Data Visualization; Kafka ; Machine Learning; Spark; Python; Elasticsearch, Kibana, Logstash.

## **1 Introduction**

Streaming data of fuel prices can be useful for a number of reasons, including: Consumer decision-making: By providing real-time or near real-time information on fuel prices at different stations, consumers can make more informed decisions about where to fill up their vehicles, potentially saving money on fuel costs. Fuel retail competition analysis: Fuel retailers can use streaming data on fuel prices to monitor

and adjust their own prices in response to competition, which can help them to stay competitive and maintain profitability. Market research: Streaming data on fuel prices can be used to study trends in the fuel market, such as changes in consumer demand, the impact of government policies, and the effects of global events on fuel prices. Economic analysis: Streaming data can be used to study the impact of fuel prices on the economy and the overall well-being of the country.

Government monitoring: Government bodies and regulatory agencies can use streaming data on fuel prices to monitor and enforce compliance with price regulations, and to assess the impact of policy decisions on fuel prices and consumer affordability. Environmental research: Streaming data on fuel prices can be used to study the relationship between fuel prices and consumption, which can help researchers to understand the impact of fuel prices on carbon emissions and air pollution. This project suggests creating a full pipeline and visual interface for fuel price tool in kafka/spark utilizing a variety of technologies, including kibana for visualization, elasticsearch for storing the data, etc.

These are just a few examples of the many open-source APIs available for streaming data processing. These technologies can handle high-throughput and low-latency data processing and analytics tasks and they can be integrated with other tools and technologies to create a complete streaming data processing pipeline.

## **2 Pipeline**

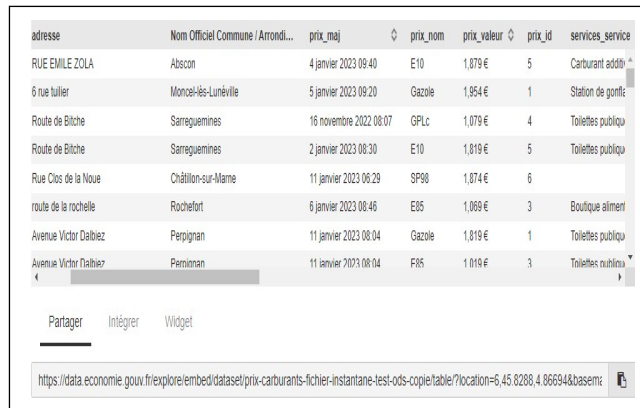
A typical pipeline for processing and analyzing streaming data can involve several stages, from data collection to the final output. Here is the work of a basic pipeline that uses Apache Kafka and Apache Spark.

### **2.1 Data Collection:**

The first step is to collect the streaming data from various sources, such as sensor networks, social media plat-

forms, or application logs. The data can be collected using various means, such as Kafka producers, REST API, or a simple script.

In this project we used an open source streaming data api diffusing data of fuel price of France every ten minutes.



adresse	Nom Officiel Commune / Arrondi...	prix_maj	prix_nom	prix_valeur	prix_id	services_service
RUE EMILE ZOLA	Abbecon	4 janvier 2023 09:40	E10	1,079 €	5	Carburant addit
6 rue Jullier	Moncel-Hés-Lunéville	5 janvier 2023 09:20	Gazole	1,054 €	1	Station de gonfl
Route de Biche	Sarreguemines	16 novembre 2022 08:07	GPLc	1,079 €	4	Toilettes public
Route de Biche	Sarreguemines	2 janvier 2023 08:30	E10	1,019 €	5	Toilettes public
Rue Clos de la Noue	Châtillon-sur-Mame	11 janvier 2023 08:29	SP98	1,074 €	6	
route de la rochelle	Rochefort	6 janvier 2023 08:46	E85	1,069 €	3	Boutique aliment
Avenue Victor Dalbiez	Perpignan	11 janvier 2023 08:04	Gazole	1,019 €	1	Toilettes public
Avenue Victor Dalbiez	Perpignan	11 janvier 2023 08:04	F05	1,019 €	3	Toilettes public

Partager Intégrer Widget

<https://data.economie.gouv.fr/explore/embed/dataset/priv-carburants-fichier-instantane-test-ots-copie/table?location=6,45,8288,4,86694&basemap=>

Fig. 1. The streaming data api

## 2.2 Data Processing:

The next stage is to process the data using a suitable technology, such as Apache Spark or Apache Flink. This can involve running complex computations, performing analytics, or training machine learning models. In our project we used:

- apache kafka: a Kafka producer is a client application that sends messages to one or more topics in a Kafka cluster. The messages in our case are the fuel price and its relative information, which are in the form of json data, are published to the topic called fuelPrice, where they can be consumed by the consumer. A Kafka producer is implemented using the Kafka Producer API, which provides a simple, high-level API for sending messages to a Kafka cluster.
- spark provides a high-level API called Resilient Distributed Datasets (RDDs), which is a fundamental data structure for Spark. RDDs are immutable, partitioned collections of objects that can be processed in parallel. Spark also provides higher-level APIs built on top of RDDs such as DataFrame and Dataset API, which are more optimized for performance and are more convenient to use than RDDs. Apache Spark can be used to process consumer files, which contain information about customers, such as their demographics, purchase history, and other relevant data. Apache Spark can be used with the PySpark library to process consumer files in Python. PySpark provides a high-level API for interacting with Spark in Python. The map function applies a lambda function to each element of the RDD and returns a new RDD with the result of the lambda function applied to each element.

```
data_json = rdd.map(lambda x: x[1]).collect()
```

Fig. 2. fig 2: mapping the data

The lambda function takes a single parameter, x, which is assumed to be a tuple, and it returns the second element of the tuple, x[1].

The result of the map operation is then collected using the collect() method. The collect method returns an array that contains all elements of the RDD. Therefore, data json variable will be a list of all the second elements of the tuples in the RDD.

## 2.3 Data Storage:

When using Elasticsearch for data storage, data is typically first ingested into Elasticsearch via an indexing process. An index is a logical namespace that maps to one or more shards and can be used to store documents. Each index is made up of one or more shards, and each shard is a self-contained index. Indices are used to store, retrieve and search for data, just like a database. To store data in Elasticsearch, one can use the Index API which indexes a JSON document. The index API adds or updates a typed JSON document in a specific index, making it searchable.

For our project an index with appropriate values is made. When data is sent to Elasticsearch, it is indexed into one and then searched, analyzed, and visualized using Elasticsearch's powerful query language and aggregations.

## 2.4 Data Visualization:

Kibana is an open-source data visualization and exploration tool that can be used in combination with Elasticsearch to analyze and visualize large volumes of data. It's a powerful tool that allows you to create interactive visualizations and dashboards to gain insights into your data.

With Kibana, we create a wide variety of visualizations, including line charts, pie charts, bar charts, and heat maps. We also use Kibana to create advanced visualizations such as maps, sunbursts, and graph visualizations. Once visualizations are created, we combined them into interactive dashboards to provide a comprehensive overview of the data. Defining the index pattern that

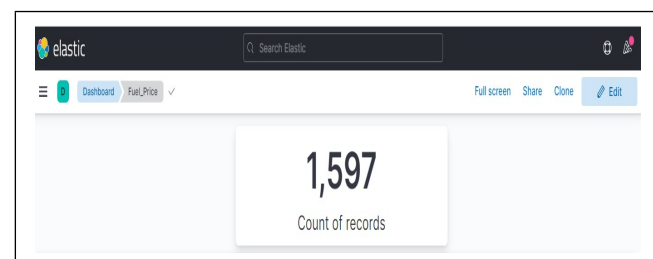


Fig. 3. fig 3: Number of data uploaded in elasticsearch

Kibana will use to connect to the Elasticsearch index 'FuelPrice'. Using the Kibana Discover feature to explore the data in the index and create new visualizations.

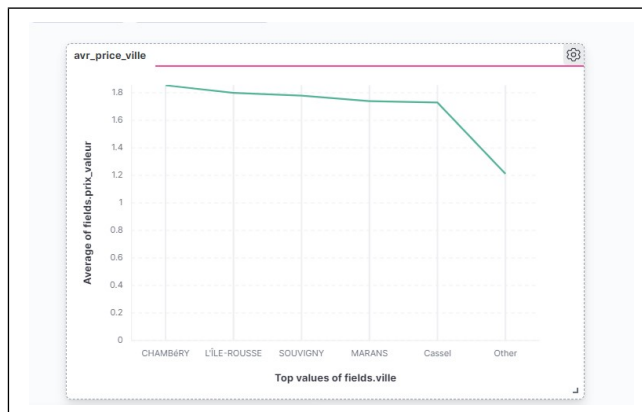


Fig. 4. fig 3: pie charts of the distribution of fuel in France

Additionally, with the help of Apache Kafka, it is possible to handle large amount of data at high throughput rates, and Spark provides an efficient and fast way to process this data and extract insights from it. On the other hand, Elasticsearch and Kibana are powerful tools for storing and visualizing the data, providing a convenient way for querying, exploring, and presenting the data.

## 4 References

- [1] <https://spark.apache.org/docs/latest/>
- [2] <https://kafka.apache.org/downloads>
- [3] <https://www.elastic.co/guide/en/elastic-stack/current/index.html>
- [4] <https://www.elastic.co/guide/en/kibana/current/index.html>

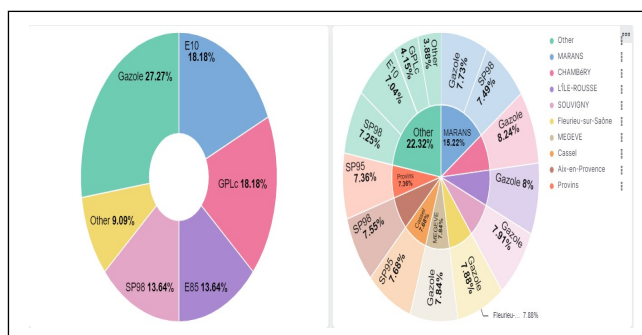


Fig. 5. fig 3: pie charts of the distribution of fuel in France

### 3 Conclusion

In conclusion, streaming fuel price data can provide valuable insights into the fuel market and can be used to make informed decisions in a variety of industries. To process and analyze this data, one can use a combination of technologies, such as Apache Kafka for collecting and ingesting the data, Apache Spark or Apache Flink for processing and analyzing the data, and Apache Elasticsearch and Kibana for storing, searching and visualizing the data.

A typical pipeline for processing this kind of data can include several stages, such as data collection, data ingestion, data processing, data storage, and data visualization. By using this pipeline, organizations can gain insights into fuel price trends, identify patterns and anomalies, and make predictions about future fuel prices.