# Traffic Status Prediction

Capstone project final report

**Matilde Silva**

Bachelors in Informatics and Computing Engineering

**U.Porto Tutor**: Carlos Baquero-Moreno
**Company Tutor**: Guilherme Soares

June 22, 2023

# Contents

# 1 Introduction

## 1.1 Project Context

In an ever-evolving world, the need to provide fast and reliable transportation information becomes increasingly necessary. Currently, there are few offers of real-time, and accurate, traffic reports, especially in Portuguese roadways. The total extent for national roadways is estimated at 12 890km, of which 23% are highways [8]. Hence, close to a quarter of all roads, are highways and, as the average traffic keeps growing with each year[9], so does the need for an algorithm to correctly determine traffic status. Furthermore, throughout the traffic road network appear traffic devices that measure car flux and/or determine vehicle type in a given location. The data collected from these devices allows for a real-time management of road incidents or inconveniences. Then, Operation Teams that monitor such events, alert road users about them.

This project was developed in relation to the Capstone Project course in FEUP's Informatics and Computer Engineering Bachelors. Nonetheless, it was mostly done in a professional work environment, in affiliation with Armis Intelligent Transport Systems, the latter providing technical expertise and support. Armis ITS manages direction of ITS Portugal, the leading structure in Intelligent Transport Systems in Portugal. This also implies a connection between Armis ITS and transport infrastructures powerhouses in Portugal, such as, Ascendi, Brisa, Infraestruturas de Portugal, S. A., Via Verde, among others[12]. Such closeness to real word applications further deepens knowledge and experience in a professional setting. In addition, access to traffic devices' real, on-time data, is a considerable advantage, on the grounds that effective traffic management requires that the practitioner works from factual information. Consequently, the results of prediction algorithms will be better suited to evaluate real-life situations.

## 1.2 Goals and expected results

The main goal of the project is to determine the traffic status continuously, based on the data gathered by the traffic monitoring devices, according to qualitative parameters. The result will be in the form of a value, either *Good, Medium Good, Medium Bad* or *Bad*. *Good* would imply that there are no major delays on a given pre-determined route, *Medium Good* means a small, not deeply impactful, delay exists on a pre-determined route, *Medium Bad* would translate into a significant delay, on a pre-determined route, and, lastly, *Bad* signals a near-stagnant traffic state, therefore a great time delay, on a pre-determined route. Artificial Intelligence techniques, such as, data analysis and machine learning will be used to relegate the results.

### 1.3 Report Structure

This report will focus on the development and timeline of the aforementioned project, touching on topics such as project methodologies, planning and development of the solution, validation, and testing of different approaches and, lastly, overall conclusions and takeaways.

## 2 Used methodology and carried out activities

### 2.1 Used methodology

The selected methodology was the waterfall model. The reason why this approach was chosen was because the necessary requirements were clear-cut and direct, also there would be no unexpected change of the possible tasks to be performed throughout the project. In addition, a fixed timeline was well-defined since the beginning of the internship.

The waterfall methodology itself is divided into the 5 following phases: Requirements, Design, Implementation, Verification and Maintenance[3]. Due to the nature of the internship, the Maintenance period would not be applicable.

### 2.2 Actors, roles and responsibilities

The actors involved in this program were I - Matilde Silva -, Guilherme Soares, Carlos Baquero-Moreno and Armis. Each played, respectively, the role of Intern, Company Tutor, U.Porto Tutor and Hosting Company.

As an Intern I was given the responsibility to develop and advance the Traffic Status project, alongside Guilherme, who supported and advised me throughout the entire process and supplied the needed information and details to further expand the proposal. Professor Baquero-Moreno also gave useful insight into techniques and tools to be used along the way. Lastly, Armis allowed me to work in an inclusive environment, providing resources and trusting me with the responsibility to carry out the proposal. I, thank each and every one of the participants, without them this truly wouldn't be possible.

## 2.3 Carried out activities

The first of many stages was to thoroughly research the topic of Traffic Engineering. Defined as the science of managing traffic flow on roads and highways, Traffic Engineering, wishes to understand how traffic flows, as a means to identify bottlenecks and develop solutions, with the greater goal of reducing congestion. To achieve their goals, traffic engineers use vast variety of resources, such as data collection and analysis, computer modeling and traffic simulations. Another ambition lies in the safety of road users; making sure the roads are well maintained, properly signaled and free of debris.[5]

Having read a few articles that gave me a better overall understanding of Traffic Engineering, a more focused search was done on Portugal's roads and traffic management. IMT, formerly IMTT, has extensive records on road use. Going more in depth, every trimester new statistics come out for each monitored highway[7]. This information is particularly useful, as the monitoring of each road must have a foresight into it's expected usage.

After a dedicated period for topic research and context familiarity, the first progress meeting with Guilherme Soares and Professor Carlos Baquero-Moreno was scheduled and carried out. During said presentation, both tutors provided useful insight, regarding tools and technologies. As Armis is a Microsoft parter company, it was suggested that I use C# as the main programming language throughout the solution development. However, after a bit of research I found that the most adequate programming languages for Machine Learning and Artificial Intelligence were R and Python[2][1].

A small sum of time was spent using the R language, but I found the learning curve rather steep, and I wasn't growing familiar with it. Consequently, a switch to Python was made, given the pre-existing knowledge.

The subsequent step involved immersing myself in a comprehensive exploration of machine learning resources and technologies. Given the vast and ever-expanding nature of this field, a sense of overwhelm was initially experienced. However, as I delved deeper into my research, a clearer path for the project gradually began to emerge.

During my research, I came to realize that machine learning can be categorized into two distinct streams: supervised and unsupervised[17]. In the realm of supervised machine learning, the availability of a target variable is crucial for training the models effectively. As we will delve into later, the database provided lacked any relevant information pertaining to traffic status, rendering supervised learning infeasible. Given the case, I was compelled to explore unsupervised machine learning, which itself can be further subdivided into the many realms, two of which are clustering and association[6]. The spotlight given to Clustering and Association is due to their technical relevance to the project's development.

Clustering focuses on grouping data points together based on their characteristics, or other specified parameters. When considering the applications of this tool, we come to realize that clustering algorithms could be a powerful tool for separating our datasets into the various traffic classifiers (*Good, Medium Good, Medium Bad* and *Bad*). In the domain of unsupervised association algorithms, generally the goal is to discover relationships hidden in larger datasets, not usually observable using traditional analysis tools. Consequently, a door is opened to better understand the correlation between the flux, velocity and other given features.

The initial list of algorithms was long, encapsuling the most common and famous of both clustering and association. However, some of those incorporated time series, while others did not. Next follows an excerpt of the drafted algorithms list:

1. Clustering Algorithms (without Time Series)
    - Agglomerative Clustering
    - BIRCH
    - DBSCAN
    - K-Means
    - Mixture of Gaussians
2. Clustering Algorithms (with Time Series)
    - K-means clustering with Dynamic Time Warping

- Hierarchical Time Series Clustering
- Time Series Motif Discovery

3. Association Algorithms (without Time Series)
   - FP-Growth algorithm
   - Sequential Pattern Mining
4. Association Algorithms (with Time Series)
   - Episode Discovery
   - Temporal Sequence Mining

Out of all these algorithms, three were chosen:

1. K-Means
2. K-means clustering with Dynamic Time Warping
3. Mixture of Gaussians

The K-Means algorithm, the most widely-used centroid-based clustering algorithm, was chosen for its versatility and generality[4]. K-means clustering with Dynamic Time Warping was chosen for partly the same reasons as K-Means, with the added benefit of time series comparison[15]. Lastly, Gaussian Mixture Models provide an insight not reachable using the K-Means algorithm - a probability associated to the cluster assignment of each data point[14].

Other algorithms were not chosen for their over-complexity, for example, BIRCH, or lack of applicability to the problem, i.e. FP-Growth algorithm.

The next phase involved creating and validating machine learning models for the specified algorithms. Having reached the appropriate conclusions, the API creation and deployment followed. More details on the development will be given later on the report.

Lastly on the list, writing the report and adding the final touches! A Gantt diagram[16] with the timeline follows:

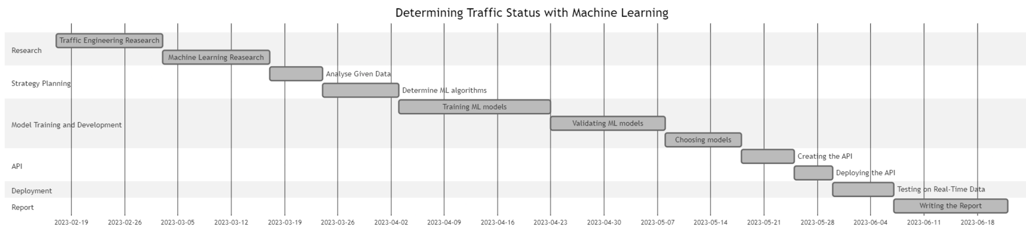


Figure 1: Determining Traffic Status with Machine Learning - Gantt Diagram

# 3 Solution Development

## 3.1 Requirements

There are two kinds of requirements, functional and non-functional. Functional requirements are defined as "*Any requirement which specifies what the system should do.*", in turn, non-functional requirements are specified as "*Any requirement that specifies how the system performs a certain function.*"[13].

The functional requirements are:

- The system should generate accurate predictions of traffic status based on the input data and trained machine learning models.
- The input data provided to the API must be feasible.
- The API should make use of efficient computation and low-latency response times to ensure real-time prediction capabilities.
- The API should return the predicted traffic status in a structured format.

The non-functional requirements are:

- The models should be trained on real historical data.
- The models should be thoroughly analysed and validated before chosen and deployed.

- The API should respond to HTTP POST requests containing form-data arguments.
- The API HTTP POST request form-data arguments should be the following: Timestamp, Flux, Velocity and Sensor ID.
- The traffic status should be one of: *Good, Medium Good, Medium Bad, Bad*

## 3.2 Solution Development

Firstly, the sensor data was stored in a database, accessible through Microsoft's SQL Server Management Studio (SSMS)[10]. The attributes were:

- MedidasCCVStatusId - Integer (unique)
- SensorCCVId - Integer
- VehicleTypeId - Integer
- Timestamp - String (YYYY-MM-DD HH:MM:SS)
- Fluxo - Integer
- Velocidade - Float
- EstadodeTrafego - String

Before the data was to be given to the algorithms, it must first be analysed and treated.

Firstly, there is no attribute that indicates to us the capacity of the road. This raises a problem for the prediction model. For example, let's take into account two roads: Road A and Road B. Road A has five lanes and Road B has only one lane. This implies that a Flux of five cars on Road A, is not necessarily a congestion; however, the same conclusion can't be made for Road B. For this reason, I found the better approach to be creating a prediction model for each individual sensor.

Next, the *VehicleTypeId* column indicates the type of vehicle that the entry refers to, i.e. there could be two distinct entries for the same *Timestamp*, but entry A only registers Motorcycles, while entry B only counts Commercial Transport Trucks. For this project, distinguishing Flux based on vehicle type is not necessary. Instead, a small computation was made, so that all entries with equal Timestamps would be replaced by a single instance, summing all corresponding *Fluxo* values and averaging the *Velocidade*. In addition, there are some sensors where the *Velocidade* attribute is NULL. We will accommodate this situation, when training the models, later on.

Afterwards, the Circular Encoding technique[11] was applied to the Timestamps, to encapsulate the circularity of time. This way, instead of a string, time is now represented by a value of *sin*and *cos*. There is a catch, nonetheless. Having two columns to represent the single feature that is time, can be misleading to some Machine Learning algorithms, as such, the *arctan* of the *sin* and *cos* values was calculated and used as the time representation. In addition, some new datasets were created with a *Weekday* column, so that a model for each day of the week, specific to a sensor, could be studied.

Lastly, the *MedidasCCVStatusId* and *EstadodeTrafego* columns were discarded, the former because the prediction models have no use for ID values and the latter because all entries were populated with NULL.

To sum up, we now have a dataset with the modified attributes:

- SensorCCVId - Integer
- Timestamp - Float (Unique)
- Fluxo - Integer
- Velocidade - Float (Only available for some sensors)

Since the *Velocidade* attribute is not available to all sensors, three subdatasets were derived for each sensor:

1. Dataset with Timestamp, Fluxo and Velocidade (when applicable)
2. Dataset with Velocidade and Fluxo (when applicable)
3. Dataset with Timestamp and Fluxo

To each sensor's three datasets we then applied the previously selected Machine Learning algorithms: *K-Means*, *K-means clustering with Dynamic Time Warping* and *Mixture of Gaussians*. Not to forget, the same was done for the weekday models, i.e. *Monday* dataset pertaining to sensor A, was parted into the three smaller datasets, which were then fed to the Machine Learning algorithms.

Promptly, we can advance to the analysis of the results for each model and corresponding datasets.

## 3.3 Model validation

The metrics used to evaluate the models were:

1. Elbow Method (only applied to *K-Means*)
2. Silhouette Score - calculated using the mean intra-cluster distance and the mean nearest-cluster distance (the higher the better)
3. Calinski-Harabaz Index - measure the distinctiveness between groups (the higher the better)
4. Davies-Bouldin Index - average similarity of each cluster with its most similar cluster (the lower the better)

Notably, these metrics, while useful to evaluate correctness, will not be the sole evaluator for the prediction models, in that, higher marks on the measurements may not directly indicate correctness of the future predictions.

In the given dataset, there were four sensor IDs, which we will evaluate individually.

### 3.3.1 Sensor 10000028

Sensor 10000028 does not have the velocity attribute, hence we only used the Dataset that contained the encoded Timestamps and corresponding Flux values to train the models. Next follow the graphs, indicating the generated clusters.

It appears obvious that, in this case, the K-Means algorithm yielded no difference in results compared to the K-Means with Dynamic Time Warping algorithm. Similarly, the Gaussian Mixtures also formed clusters based on the *Fluxo* attribute.

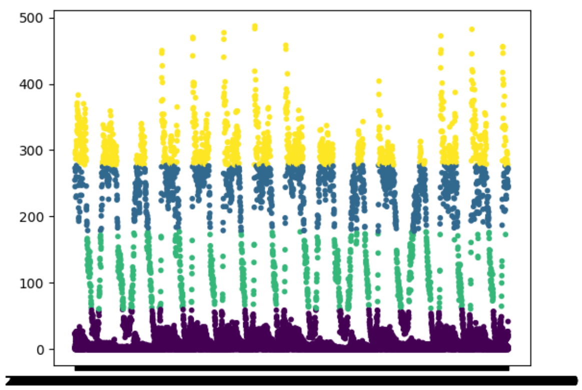


Figure 2: K-Means algorithm. Flux on the Y axis, Time on the X axis - Sensor 10000028

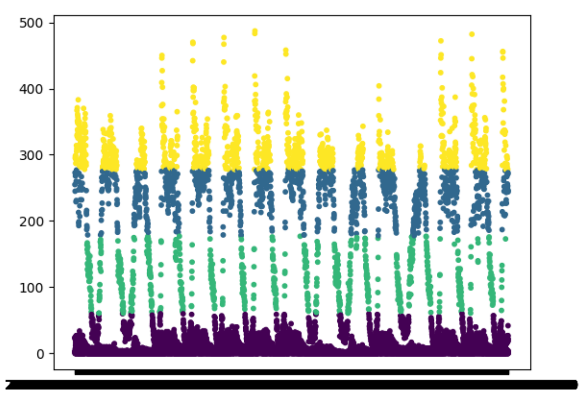


Figure 3: K-means with DTW algorithm. Flux on the Y axis, Time on the X axis - Sensor 10000028

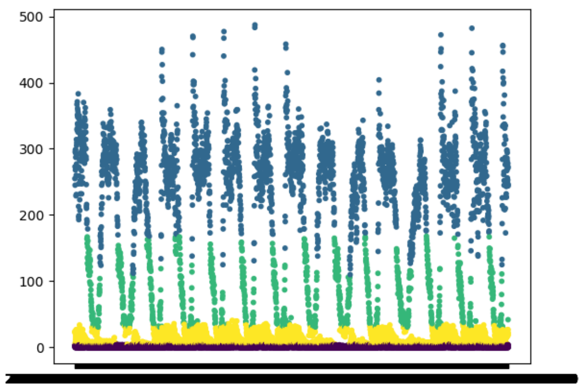


Figure 4: Gaussian Mixture Algorithm. Flux on the Y axis, Time on the X axis - Sensor 10000028

Next follows the table with the performance measures:

| | K-Means | K-Means with DTW | Gaussian Mixtures |
|---|---|---|---|
| Silhouette Score | 0.6916 | 0.8309 | 0.5654 |
| Calinski-Harabaz Index | 308866.22 | 178713.17 | 85819.73 |
| Davies-Bouldin Index | 0.5315 | 0.5073 | 0.6181 |

Based on the values and graphics, none of the algorithms stand out as better or worse, even so, we will choose the K-Means with DTW as the most suitable model. Extraneous, the missing *Velocidade* column appears to be a great fault in the model training.

### 3.3.2 Sensor 10000029

Sensor 10000029 also does not have the velocity attribute. Here are the graphs:

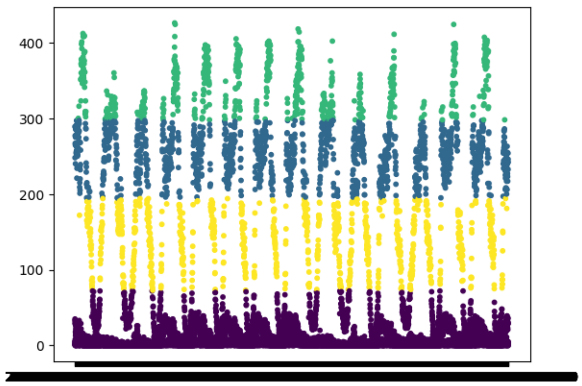


Figure 5: K-Means algorithm. Flux on the Y axis, Time on the X axis - Sensor 10000029

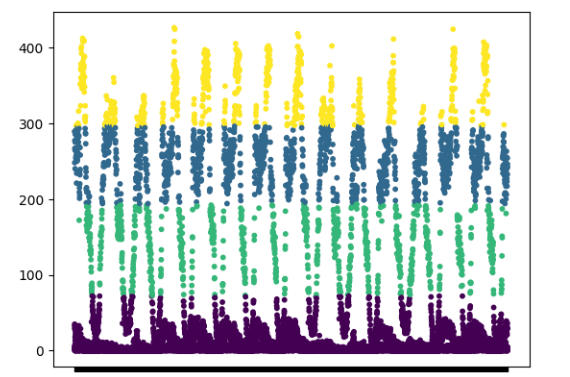


Figure 6: K-means with DTW algorithm. Flux on the Y axis, Time on the X axis - Sensor 10000029

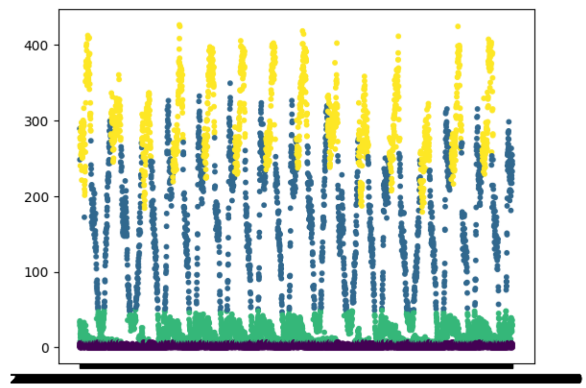


Figure 7: Gaussian Mixture Algorithm. Flux on the Y axis, Time on the X axis - Sensor 10000029

And the performance metrics:

| | K-Means | K-Means with DTW | Gaussian Mixtures |
|---|---|---|---|
| Silhouette Score | 0.7177 | 0.8468 | 0.5820 |
| Calinski-Harabaz Index | 335612.24 | 173860.19 | 58354.61 |
| Davies-Bouldin Index | 0.4975 | 0.4628 | 0.7123 |

Commensurate to sensor 10000028, the results between models are not dissimilar. Again, K-Means with DTW takes the lead as the most suitable model.

#### 3.3.3 Sensor 10010172

Sensor 10010172 does have the velocity attribute, meaning we have 3 datasets to take advantage of. We will dive deeper into the Dataset with Timestamp, Flux and Velocity, out of the three, since it yielded the most interesting results.

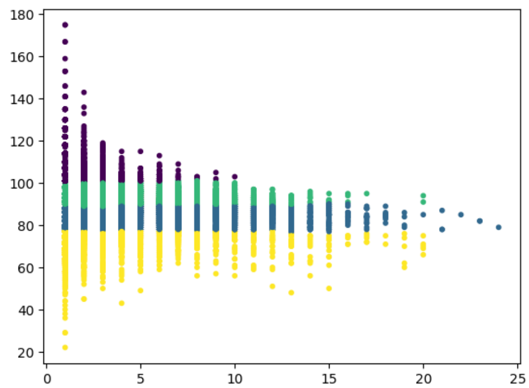


Figure 8: K-Means algorithm. Velocity on the Y axis, Flux on the X axis - Sensor 10010172

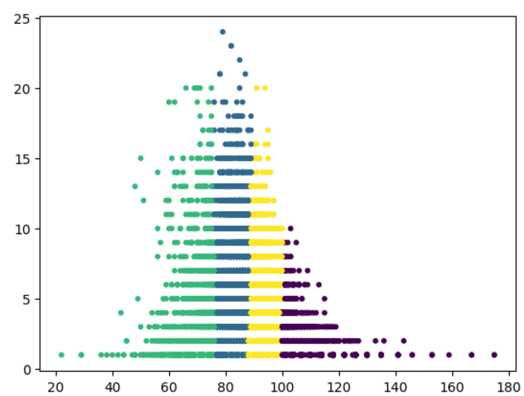


Figure 9: K-means with DTW algorithm. Flux on the Y axis, Velocity on the X axis - Sensor 10010172

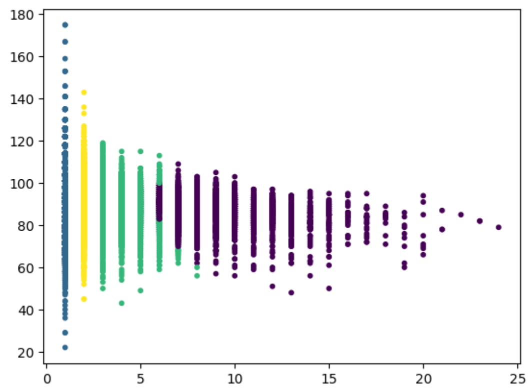


Figure 10: Gaussian Mixture Algorithm. Velocity on the Y axis, Flux on the X axis - Sensor 10010172

Also, the metric results:

| | K-Means | K-Means with DTW | Gaussian Mixtures |
|---|---|---|---|
| Silhouette Score | 0.3042 | 0.3642 | -0.0176 |
| Calinski-Harabaz Index | 15020.91 | 17325.82 | 537.76 |
| Davies-Bouldin Index | 0.9354 | 0.8707 | 9.224 |

The K-Means models seamed to cluster the data based on the velocity values, which results in a no better division in clusters, than that of a simple if statement. The gaussian model, however, did make some distinctions between equal values of velocity, based on the flux. These are our most promising results yet. For now, Mixture of Gaussians takes the front!

#### 3.3.4 Sensor 10010173

Like sensor 10010172, sensor 10010173 dataset's includes the velocity column. We will use the same approach as sensor 10010172.

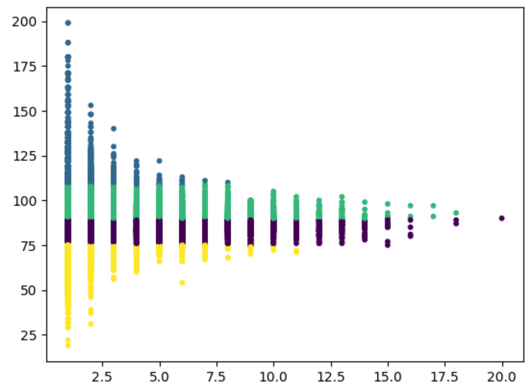


Figure 11: K-Means algorithm. Velocity on the Y axis, Flux on the X axis - Sensor 10010173

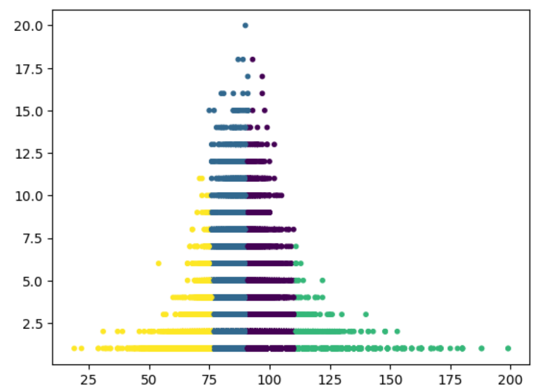


Figure 12: K-means with DTW algorithm. Flux on the Y axis, Velocity on the X axis - Sensor 10010173

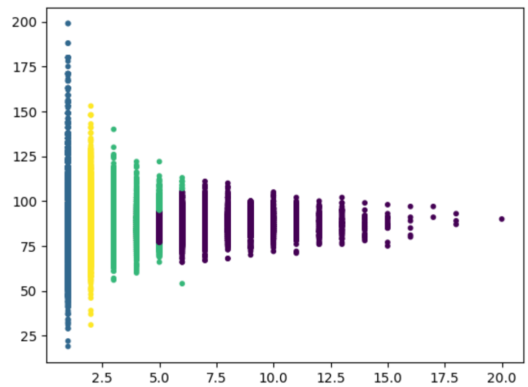


Figure 13: Gaussian Mixture Algorithm. Velocity on the Y axis, Flux on the X axis - Sensor 10010173

Lastly, the performances:

| | K-Means | K-Means with DTW | Gaussian Mixtures |
|---|---|---|---|
| Silhouette Score | 0.3193 | 0.4064 | -0.0789 |
| Calinski-Harabaz Index | 24572.14 | 23085.29 | 433.59 |
| Davies-Bouldin Index | 0.8494 | 0.7370 | 13.624 |

Like before, K-Means, both with and without Dynamic Time Warping, didn't do much. Hence, based on all the performances, the Gaussian Mixtures is the most comprehensive Machine Learning algorithm. Even though, it may not have been a perfect fit for sensors 10000028 and 10000029, the difference between K-Means, in this case, is not immeasurable.

Hence, for the sake of simplicity, we will use the Mixtures of Gaussian Models, on all sensors.

#### 3.3.5 Weekday models

The data that I was given was spread out over two weeks, thus, the weekday models contained only 2 days worth of data. The models did not perform any better with the separated weekdays, thus the results were not worth the extra computations, training and storage. As a result, the idea for a model for each day of the week was scraped.

## 3.4 Final Product

Now, the culmination of all the described work lays in a single Docker container. Said container was deployed using Azure. The API rests on the following domain: `https://traffic-status-api.azurewebsites.net/`.

To get the Traffic Status, one must send a HTTP POST request with the following form-data arguments:

- Timestamp
- Fluxo
- Velocidade
- SensorID

To facilitate this task, we can use the Postman application:

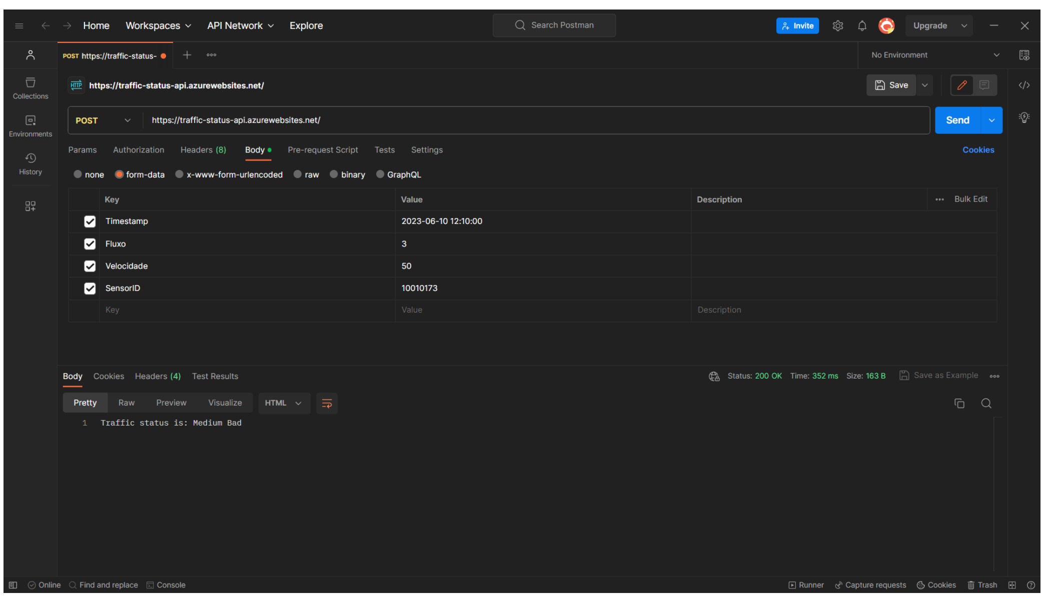


Figure 14: HTTP POST request example, using Postman.

The response will be string of the format *Traffic status is: {traffic status}*

The architecture diagram follows:

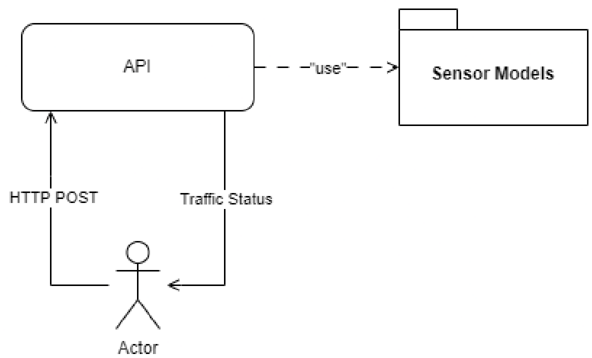


Figure 15: API architecture

## 3.5 Technologies

In this project, the used technologies were:

- Python - most importantly, SKLearn Library
- Jupyter Notebook
- Microsoft's SQL Server Management Studio (SSMS)
- Flask
- Postman
- Azure
- Docker

# 4 Conclusions

## 4.1 Reached results

The objective of this project was to create a predictor for Traffic Status, using Machine Learning. In the beginning, I expected the result of this project to be all or nothing. The prediction model would either be accurate to a fault or it would be rendered useless. Now, more than ever, I realize how unfounded and unrealistic my expectations were.

Even though the predictions might not fall on the 100% accuracy mark, I am satisfied with the results of this project. I believe I have created a sturdy knowledge base, which will allow this project to blossom and thrive!

## 4.2 Lessons learned

Throughout the course of this internship, I have learned so much. At the start, I couldn't even imagine how a professional work environment functioned. I am truly grateful to have had the opportunity to work with the support of Armis and all those in the Intelligent Transport Systems department. I couldn't have asked for a more kind and knowledgeable team! Additionally, I have discovered a new area of interest, Intelligent Transports. I hope that, in the future, I am able to take the steps to satisfy my curiosity regarding this theme. Lastly, I am also much more involved in the area of Artificial Intelligence, which, in recent times, is rapidly advancing.

## 4.3 Future work

Some improvements could be made to the project, to better guarantee trustworthy predictions. The first possibility: use more data to train the models, if available. Even though there were plenty of entries, more than two weeks of traffic information could prove extremely beneficial to the functioning of the model. Furthermore, training weekday models with the additional database entries, to analyse the possibility, and usefulness, of a prediction model for each day of the week.

Moreover, as mentioned before, the information regarding the capacity of each road was not available. The road's constant capacity is useful to be aware of how many cars can be travelling simultaneously on it. For example, an immensely used highway like the A1, will have a larger capacity than a less frequented national road. If it were possible to explore said values, one could experiment with a single unique model, suitable to all roads (a proportion between flux and road capacity, perhaps).

Lastly, a form interface on the API website, to facilitate its use and make it more user-friendly. In conjunction to this, validating input data, before the HTTP POST request is made, thus avoiding a crash.

# References


[1] Coursera. Python or r for data analysis. https://www.coursera.org/articles/python-or-r-for-data-analysis, 2023. [Online; Accessed on 19 May 2023].

[2] Datacamp. R and python for data science. https://www.datacamp.com/blog/python-vs-r-for-data-science-whats-the-difference, 2022. [Online; Accessed on 19 May 2023].

[3] Forbes. Waterfall methodology. https://www.forbes.com/advisor/business/what-is-waterfall-methodology/, 2022. [Online; Accessed on 19 May 2023].

[4] FreeCodeCamp. 8 clustering algorithms in machine learning that all data scientists should know. https://tinyurl.com/mutcwpmc, 2020. [Online; Accessed on 16 June 2023].

[5] Greenlight. What is traffic engineering? https://www.greenlighttrafficengineering.com/what-is-traffic-engineering/. [Online; Accessed on 16 June 2023].

[6] IBM. What is unsupervised learning? https://www.ibm.com/topics/unsupervised-learning. [Online; Accessed on 16 June 2023].

[7] IMT. Relatórios de tráfego. https://www.imt-ip.pt/sites/IMTT/Portugues/InfraestruturasRodoviarias/RedeRodoviaria/Paginas/Relatorios.aspx. [Online; Accessed on 16 June 2023].

[8] IMT. Highway percentage. https://www.imt-ip.pt/sites/IMTT/Portugues/Planeamento/DocumentosdeReferencia/

Planeamento-Acessibilidades-Gestao-Viaria/Documents/01EngTrafego.pdf/, 2008. [Online; Accessed on 19 May 2023].

[9] IMT. Usage of roads. https://www.imt-ip.pt/sites/IMTT/Portugues/InfraestruturasRodoviarias/RedeRodoviaria/Relatrios/Relat%c3%b3rio%20de%20Tr%c3%a1fego%20-%203%c2%ba%20Trimestre%20de%202022.pdf/, 2022. [Online; Accessed on 19 May 2023].

[10] Microsoft. What is sql server management studio (ssms)? https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16, 2023. [Online; Accessed on 18 June 2023].

[11] NVIDIA. Three approaches to encoding time information as features for ml models. https://tinyurl.com/43bnfnp2, 2022. [Online; Accessed on 18 June 2023].

[12] ITS Portugal. Associtesof its portugal. https://www.its-portugal.com/associados/, 2023. [Online; Accessed on 19 May 2023].

[13] Reqtest. Why is the difference between functional and non-functional requirements important? https://reqtest.com/en/knowledgebase/functional-vs-non-functional-requirements/, 2012. [Online; Accessed on 16 June 2023].

[14] TowardsDataScience. How to apply k-means clustering to time series data. https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95, 2019. [Online; Accessed on 16 June 2023].

[15] TowardsDataScience. How to apply k-means clustering to time series data. https://www.datacamp.com/blog/python-vs-r-for-data-science-whats-the-difference, 2020. [Online; Accessed on 16 June 2023].

[16] wikipedia. Gantt chart. https://en.wikipedia.org/wiki/Gantt_chart, 2023. [Online; Accessed on 3 March 2023].

[17] Wikipedia. Machine learning. https://en.wikipedia.org/wiki/Machine_learning#Approaches, 2023. [Online; Accessed on 16 June 2023].