Predicting Delays on Public Transportation: A Machine Learning Application on the Parisian RER and Metro

Arina Agaronyan and Grace Beyoko

January 2025

Contents

1 Introduction

As regular users of the Parisian public transportation system, we frequently find ourselves stuck in the metro or RER, frustrated by delays that disrupt our daily routines. These unexpected interruptions often leave us wondering, "If only there were a way to predict this delay in advance". This frustration, that we are sure many have felt, became the inspiration for this project. This study attempts to explore the potential of machine learning in addressing one of the most common inconveniences faced by urban commuters.

Ideally, we imagine a future where passengers could receive in advance notifications about the likelihood of a delay at given time for a given stops, enabling users to make informed decisions about their travel plans. This paper represents the initial step toward that goal. Specifically, it focuses on developing models to predict whether a train at a given station will be on time or delayed, providing a foundation for more sophisticated applications in the future.

In the following sections, the sources and types of data used will be discussed, along with descriptive statistics. We then detail the machine learning algorithms employed in this study, explaining their methodologies and evaluating their performance. Finally, we conclude by examining the implications of our findings and outlining potential improvements to our predictive models.

2 Data

2.1 Sources and types

The primary source of data for this study is the set of APIs provided by the Île-de-France Mobilités website. Specifically, we used :

- Next Departures (Île-de-France Mobilités platform) - Unitary query: Offers data on incoming trains at specific stops. Provides train arrival and departure data from the previous hour for each stop point. This API includes details such as the train's direction, scheduled and actual arrival times, and departure status (on time, delayed, or cancelled)

- Messages displayed on the screens (Île-de-France Mobilités platform): Contains the messages displayed on station screens. Provides general information about ongoing disruptions in the metro and RER networks.

- Île-de-France Mobilités Calculator - Traffic Messages (v2) API: Provides diverse information about overall traffic status.

In addition to the APIs provided by Île-de-France Mobilités, we also used hourly Navigo validation per stops data, weather and gas prices data.

Once we identified our sources, we set the data collection code to run every hour and thus periodically log the data consistently over the set period from the 2nd November 2024 to the 22nd January 2025. Additionally, within the API collection code, we included a function to calculate the time difference between the expected arrival/departure time and the real arrival/departure time, thus allowing us to know the duration of delays. We set the code to assign a train status as delayed if the

time difference exceeds 3 minutes for the RER - we were unable to do the same for the metro data as it did not include scheduled times. Due to the frequency and high volume of the data, the collection csv files had to be split in mid-December and later merged back together in the cleaning.

These were later filtered and combined with the 15 minutely weather data collected from the open-meteo API[1], and monthly average Gas price data for gazole, E10, octane 95 and 98[2]. Additionally, we included the national holidays data, with both school and bank holidays[3].

Finally, we were left with two main datasets : **merged_RER** and **merged_metro**.The full RER dataset has 12,215,975 rows and 36 columns. Some of the columns were constructed temporarily in order to facilitate merging our secondly data with daily data such as holiday dates – the full list of remaining columns contains the following:

- **timestamp**: representing at which time the data was logged.
- **stop_reference**: the code for the stop.
- **stop_name**: the name of the stop.
- **line_ref**: the code of the line.
- **destination_name**: the destination of the given train.
- **departure_status**: whether the train was on time, delayed, or cancelled.
- **scheduled_arrival**: the scheduled time of arrival.
- **real_arrival**: the actual time of arrival.
- **scheduled_departure**: the scheduled time of departure.
- **real_departure**: the actual time of departure.
- **arrival_difference**: the difference between the scheduled arrival and actual arrival in minutes.
- **departure_difference**: the difference between the scheduled departure and actual departure in minutes.
- **ArRTown**: the city in which the stop is located.
- **nearest_datetime**: the nearest timestamp.
- **95**: the octane 95 mothly average price.
- **98**: the octane 98 mothly average price.
- **E10**: the E10 mothly average price.
- **gazole**: the gazole mothly average price.

[1] https://open-meteo.com/en/docs

[2] https://www.insee.fr/fr/statistiques/series/103157792?PERIODICITE=2224021&PRIX_CONSO=112977865+2409120+2409121+2409122+2409123

[3] outlinedhttps://public.opendatasoft.com/explore/dataset/vacances-scolaires-par-zone/export/?sort=-date

- **temperature_2m**: air temperature at 2 meters above ground.
- **precipitation**: total precipitation (rain, showers, snow) sum of the preceding 15 minutes
- **rain**: rain from large scale weather systems of the preceding 15 minutes in millimeters
- **snowfall**: snowfall amount of the preceding 15 minutes in centimeters.
- **wind_speed_10m**: wind speed at 10 meters above ground (standard level).
- **wind_gusts_10m**: gusts at 10 meters above ground as a maximum of the preceding 15 minutes.
- **visibility**: viewing distance in meters - nfluenced by low clouds, humidity and aerosols.
- **is_day**: a binary variable (0 for night time).
- **Date**: the date, constructed to merge with daily data instead of minutely.
- **day_of_week**: a number representing the day of the week (0 for Monday).
- **day_type**: the type of day.
- **is_bank_holiday**: a binary variable taking the value 1 if the day is a bank holiday.
- **is_holiday**: a binary variable taking the value 1 if the day is within a holiday season.
- **saturday**: a binary variable taking the value 1 if the day is Saturday.
- **sunday**: a binary variable taking the value 1 if the day is Sunday.
- **is_weekend**: a binary variable taking the value 1 if the day is a weekend.
- **is_weekend_or_bank_holiday**: a binary variable taking the value 1 if the day is a bank holiday or within a holiday season.
- **hour**: a number from 0 to 23 representing the hour of the day.

The metro dataset is constructed in the same manner but does not contain gas prices, scheduled departure time, scheduled arrival, arrival difference and departure difference. Our full metro data is made of 4,636,499 rows.

Due to the full datasets being unfortunately too large, we are not able to include them along this report. For reproduction purposes, we made the full dataset available on a Google Drive[4] to be downloaded and placed within the Collected_Data folder.

[4] `https://drive.google.com/drive/folders/1aTlL6tb3JCpl8b6GvRI-MgBJ0cEMES_G?usp=sharing`

3 Descriptive Statistics

In this section, we analyse statistics regarding the distributions of delays in the metro and RER systems. For better readability, below is the code used for the plots:

```
For RER :

'C01743': 'RER B',
'C01742': 'RER A',
'C01727': 'RER C',
'C01728': 'RER D',
'C01729': 'RER E'

For metro :
'C01371':  Metro_1
'C01372': Metro_2
'C01373': Metro_3
'C01374': Metro_4
'C01375': Metro_5
'C01376': Metro_6
'C01377': Metro_7
'C01378': Metro_8
'C01379': Metro_9
'C01380': Metro_10
'C01381': Metro_11
'C01382': Metro_12
'C01383': Metro_13
'C01384': Metro_14
'C01386': Metro_3B
'C01387': Metro_7B
```

3.1 Visual analysis

Here, we can see that on the metro the amount of delays tends to fluctuate between 400 to 1200 delays per days.

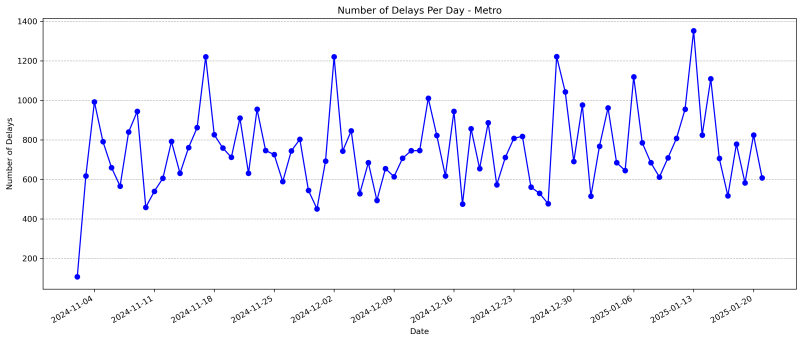


Figure 1: Average number of delay per day - Metro

The graph below provides additional insights. Specifically, it reveals that the

beginning of the week and weekends are the periods most prone to delays, with Mondays showing the highest average number of delays.

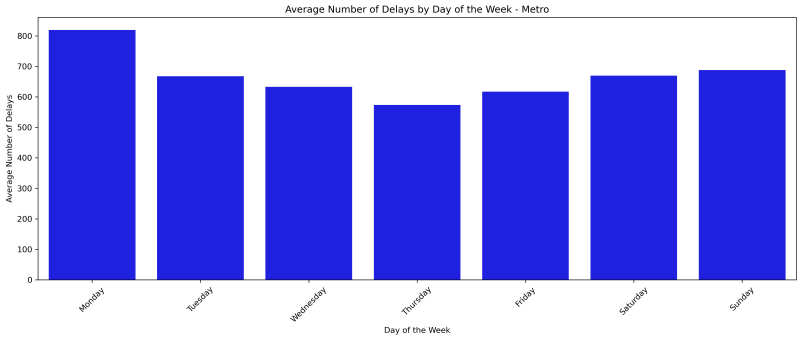


Figure 2: Average number of delays by days of the week - Metro.

Figure 3 ranks the metro lines by the number of delays. From this, we observe that Line 8 experiences the most delays, while Line 3B has almost no recorded delays (to be exact, there were 3 delays recorded over the 2.5 month period).

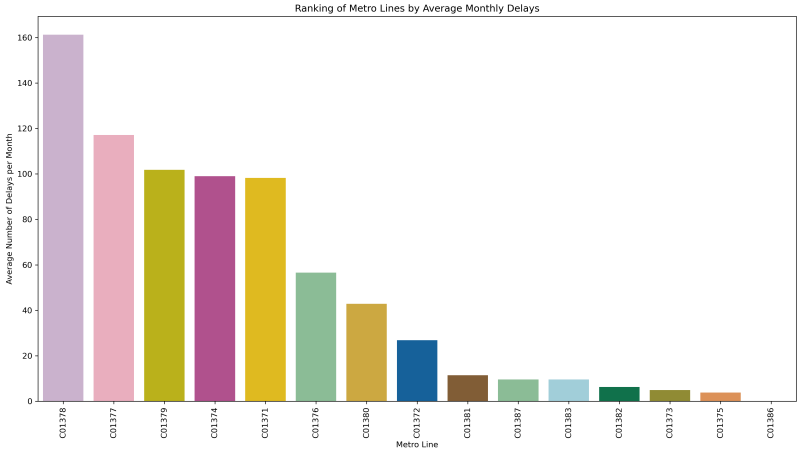


Figure 3: Ranking of metro lines by amount of delays

Figure 4 illustrates the average delay duration based on the day of the week. Surprisingly, the weekend appears to be the period when the RER system is the most efficient.

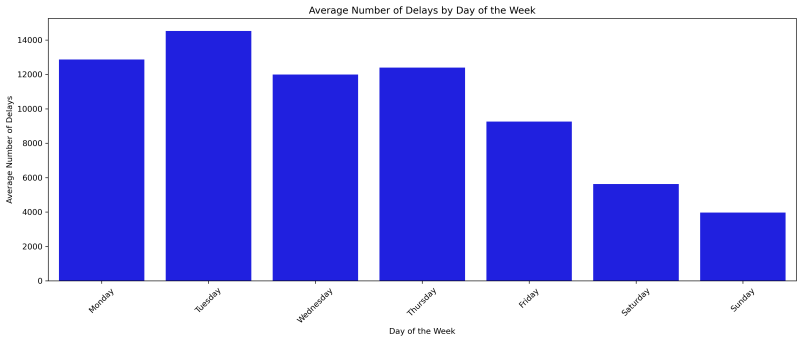


Figure 4: Average delay duration per day of the week - RER.

In Figure 5, we rank the RER lines by their proportion of delays. This allows us to see that RER B has the highest proportion of delays, while RER E (not included in the predictive analysis) has the lowest proportion of delays.

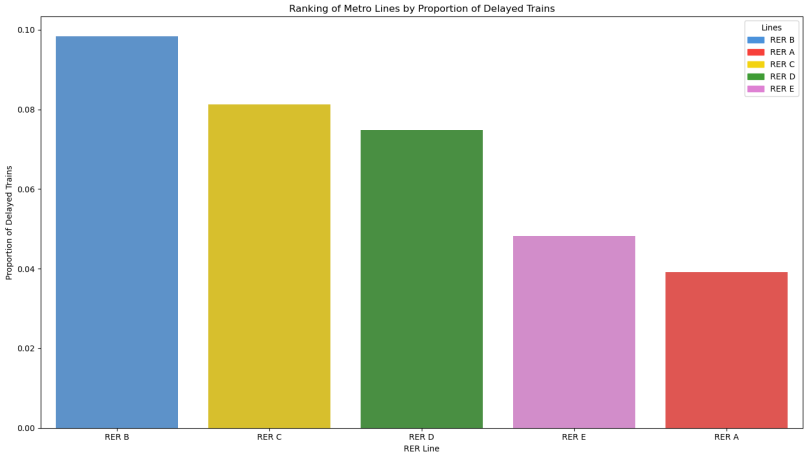


Figure 5: Ranking of RER lines by proportion of delayed trains

3.2 Variables of interest

To ease our processing and reduce the computational power needed, we decided to only focus on a selected number of stops. To avoid bias in the dataset and ensure a diverse range of stops, we chose to analyse automated and non-automated lines, as well as making a distinction between stops that connect with other lines. For RER we also made sure to include stops that are outside of Paris.

To be clear, for the RER we picked Massy - Palaiseau (RER B) as the outside-Paris connection stop, Gare de Buno Gironville (RER D) as the outside-Paris stand-alone stop, Avenue du Président Kennedy (RER C) as the in-Paris standalone stop, and Châtelet - Les Halles (RER A) as the in-Paris connection stop.

For the metro we picked Saint-Lazare (M12) as the non-automated connection stop, Blanche (M2) as the non-automated stand-alone stop, Charles de Gaulle Etoile (M1) as the automated connection stop, and St Germain-des-Près (M4) as the automated stand-alone stop.

In Figure 6, we can observe that at the Avenue du Président Kennedy station, the longest delays in terms of duration also occur during rush hours.

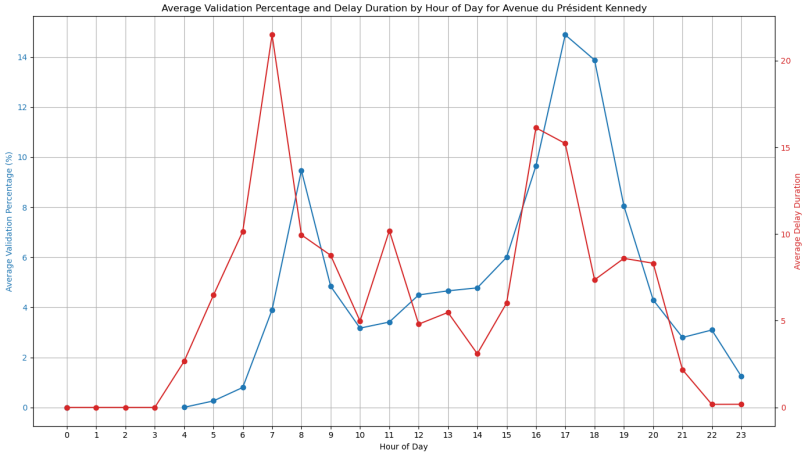


Figure 6: Average Traffic and Delay Duration by hour - Avenue du Président Kennedy

In Figure 7, we can see that at the Saint Lazare station, the trend of number of delays per hours mostly follows the traffic trend.

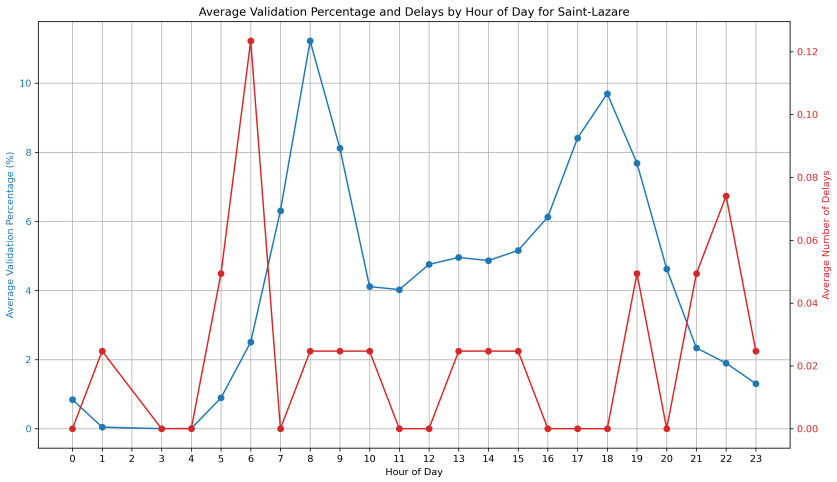
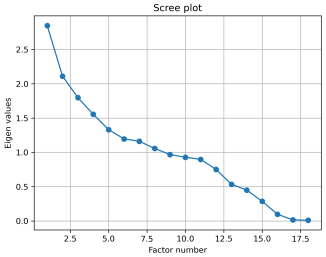


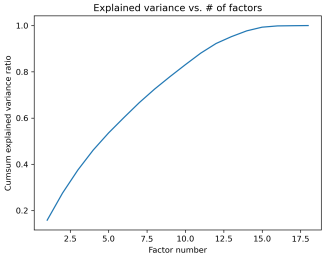
Figure 7: Average Traffic and Number of Delay by hour - Saint Lazare.

3.3 Principle Component Analysis (PCA)

Prior to executing the predictive models, we decided to analysis the the independent variables in the datasets. First, we selected the non-categorical numerical variables and standardised them due to their differing units and scales. Then, we applied a PCA with fairly similar results across stations - around 10 factors for over 80% of explainable variance.



(a) Scree Plot.



(b) Explained Variance.

Figure 8: Scree Plot and Cumulative Sum Plot for St Germain-des-Près.

4 Machine Learning Models

In this section, we chose to evaluate two models to compare their performances. Specifically, we opted to test a simpler model, logistic regression, against a more advanced model, random forest, to assess their respective strengths and suitability for our task.

4.1 Pre-processing

Before training our model, it was necessary to preprocess the variables to ensure they were suitable for regression analysis. In fact, logistic regression in particular, requires

all variables to be encoded as integers to function correctly. This step was crucial in ensuring that the model could process the data without encountering errors or inaccuracies.

Given the nature of our study, the data points are heavily influenced by time. It is reasonable to expect that delays are more frequent during peak hours, such as morning and evening rush periods. Consequently, it was important for the model to capture the cyclical nature of time across various scales (months, weeks, days, hours, minutes, and seconds). In other words, we needed the model to recognise that times like 23:50 and 00:10 are at the same distance from midnight. To account for that, we applied cyclical encoding to the time-related variables.

Next, we rescaled all non-binary variables to standardise their ranges, ensuring that variables with larger scales did not disproportionately influence the model's performance. Additionally, we transformed our target variable into binary format: assigning a value of 1 to indicate a delay and 0 for on time. For the stops on the RER C and D, lines we also transformed the cancelled status as delayed.

Finally, we defined our in-sample period as the time-frame from November 2, 2024, to January 12, 2025, while our out-of-sample period extends from January 13, 2025, to January 19, 2025.

These preprocessing steps ensured that the data was properly prepared for training, improving the model's ability to learn patterns and make accurate predictions.

4.2 Logistic Regression

The results of our Logistic regression are not as intended. Indeed, the model gives very high accuracy but never predicts delays. Consistently across all lines, metro and RER, the logistic regression always predicts an on time status for all observations. This is reasonable due to the distribution of the predictors; however, it is an unsatisfying result for a model.

4.3 Random Forest

4.3.1 Prediction for the Metro

For the metro lines, for St Germain-des-Près and Charles de Gaulle Etoile, the Random Forest regression resulted in fairly successful predictions, while for Blanche and more so for Saint-Lazare, the number of observations in one week were not enough for meaningful results, and even after resampling over 2 and 3 weeks respectively (adjusting the in- and out-of-sample data ranges), there were barely any 'delayed' classifications predicted.

Specifically, the automated lines (M1 and M4) had the best predictions, with 20 of the 21 delays correctly predicted.

	Predicted delay	Predicted onTime
Actual delay	20	1
Actual onTime	0	1101

Table 1: Charles de Gaulle Etoile Confusion Matrix.

4.3.2 Prediction for the RER

Similarly, the RER predictions were mostly unsatisfying, with at most 6 delays out of 126 predicted correctly and some stops remaining at 0 delays predicted even after resampling. Occasionally, the accuracy of the test data prediction was higher than that of the training data, likely due to the train/test data imbalance of 2 months compared to 1 week.

	Predicted onTime	Predicted delay
Actual onTime	4033	0
Actual delay	43	0

Table 2: Gare de Buno Gironville Confusion Matrix.

5 Limitations

After examining our data, we observed that the class of interest (i.e delays) made a very small proportion of the overall dataset. On average, delayed trains represented less than 5% of the total entries, with this proportion being even smaller for certain lines. This imbalance resulted in a significant challenge for our predictions. Due to the skewed nature of the dataset, our model struggled to learn the patterns associated with delayed trains effectively. Most of the time, the model defaulted to predicting the majority class, which in our case was "on time". While this resulted in consistently high accuracy (over 95%), it was misleading because the model failed to meet our primary objective. Our focus was on accurately predicting delays, as the value of correctly identifying a delayed train far outweighed the gain from correctly predicting an on time train. Thus, the high accuracy masked the models inability to address our core objective.

In addition to the imbalance issue, we encountered other challenges during data preprocessing. Several datasets, such as the "travaux" information in the line_reports datasets, which contained information about ongoing construction work on the subway and RER, were in text format. Unfortunately, due to a lack of expertise in natural language processing, we were not able to process language in a way that would convey the critical information of this dataset. As a result, critical information that could have improved the model's predictions was left unused. Similar difficulties arose with datasets that detailed the causes of delays, further limiting the insights we could derive from the available data.

Another significant problem was the large size of the dataset. The high number of entries made it challenging to compute summary statistics and execute certain functions without encountering technical issues. Frequently, the kernel would crash during processing, interrupting our workflow. Furthermore, we had difficulty in freely using functions, which exacerbated the problem as certain variables remained in memory and were unintentionally reused when they should have been reset. This increased the likelihood of code-related errors, which in turn raised concerns about potential overfitting and biased results.

6 Potential Improvements

6.1 Definition of delays

To try to resolve the unbalanced nature of our data, we first opted to change the definition of delays. In fact, the RATP defined a train as delayed when the delay duration exceeds 5 minutes. As mentioned previously, we re-classified a train as delayed when the delay exceeds 3 minutes. By doing so, we were able to increase the number of delays in our data (reaching almost 10% of the total entries in the most fortunate cases). However, even with this change, the result did not vary significantly.

6.2 Resampling

As a second approach, we explored resampling techniques to reduce the class imbalance. Specifically, we employed the Synthetic Minority Oversampling Technique (SMOTE) to "oversample" the minority class (delays). This method generates synthetic samples to increase the representation of the minority class while preserving the temporal patterns in the data. By applying SMOTE, we were able to create a more balanced dataset. However, similar to the first approach, this method did not result in a substantial improvement in model performance.

7 Conclusion

To conclude, our study aimed to predict delays in the Parisian metro and RER system using machine learning techniques. Despite the challenges faced, including severe class imbalance, unprocessed textual data, and computational constraints, we were able to test and compare the performance of logistic regression and random forest models. While our models showed high accuracy, their inability to effectively predict delays, the primary focus of this study, highlighted the limitations of our approach.

Efforts to address class imbalance, such as redefining delays and using SMOTE for resampling, helped to create a more balanced dataset but did not significantly improve model performance. Furthermore, technical challenges related to dataset size and kernel crashes restricted our ability to explore more complex models.

Future work could focus on improving data preprocessing, particularly for text-based datasets. Additionally, addressing computational constraints, perhaps by using cloud-based resources, would enable more robust analysis and experimentation.

While the results of this study fell short of achieving highly accurate delay predictions, it serves as a valuable step toward understanding the complexities of seemingly unpredictable metro delays and exploring the potential of machine learning in this context.